UNIVERSITY OF MELBOURNE

DOCTORAL THESIS

Significant Revision Identification between Revised Texts in a Multi-Author Environment

---

Author:
Ping Ping TAN
ORCID: 0000-0003-3798-0199

Supervisor:
Karin VERSPOOR,
Tim MILLER

A thesis submitted in total fulfilment of the requirements for the degree of Doctor of Philosophy

in the

School of Computing and Information Systems

June, 2019

UNIVERSITY OF MELBOURNE

Abstract

School of Computing and Information Systems

Doctor of Philosophy

Significant Revision Identification between Revised Texts in a Multi-Author Environment

by Ping Ping TAN

Despite advancement in collaborative writing tools, the track changes capability remains limited to highlighting syntactic changes, with authors still required to manually read through each of the revisions. We envision a collaborative authoring system where an author could accept all minor edits first and then focus on the substantial changes. The primary goal of this thesis is to develop a computational framework for significant revision identification where paraphrase approaches cannot fully support such identification. An existing taxonomy of revision analysis categorises revisions to surface (i.e. no meaning) and text-base (i.e. meaning) changes, with further categorisation of surface change to formal changes and meaning preserving changes, while text-base change is sub-divided to micro-structure and macro-structure changes. However, the taxonomy lacks details for computational modelling. Through examination of the works in the domain of psycho-linguistics, introspective analysis and feedback from both authors and non-authors on what constitute significant revisions, a conceptual framework for significant revision identification is proposed. An inter-rater agreement of alpha Krippendorff = 0.745 was obtained for the annotation between the authors and non-authors. The core concept of our proposed approach is bi-directional textual entailment assessment. We demonstrated that this concept is computationally feasible by relying on existing textual entailment systems. Our proposed approach is more accurate (micro-averaged F_1 = 0.541) compared to several baseline approaches based on edit distance, which are similar to the current track changes capability built in most of the word processors. Computationally identifying significant revisions between two versions of a text document has the potential to improve the revision process in a multi-author environment when multiple revisions are done by different authors.

Declaration of Authorship

I, Ping Ping TAN, declare that this thesis titled, "Significant Revision Identification between Revised Texts in a Multi-Author Environment" and the work presented in it are my own. I confirm that:

- due acknowledgement has been made in the text to all other material used; and
- the thesis is fewer than the 100 000 word limit in length, exclusive of tables, maps, bibliographies and appendices.

Signed: *Tan Ping Ping*

Date: January 7, 2020

Preface

The following peer-reviewed publications were published in the candidature:

Tan, P. P. , Verspoor, K. & Miller, T. (2015). Structural alignment as the basis to improve significant change detection in versioned sentences. In Proceedings of the Australasian Language Technology Association Workshop 2015 (pp. 101-109).

Tan, P. P. , Verspoor, K. & Miller, T. (2016). Rev at SEMEVAL-2016 Task 2: Aligning chunks by lexical, part of speech and semantic equivalence. In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016) (pp. 777-782).

Acknowledgements

My PhD journey had been an amazing journey and I forever grateful to everyone who had been on this journey with me. To my dearest supervisors Karin Verspoor and Tim Miller, thank you very much for your patience and guidance. I am thankful to all the annotators: Philip, Liz, Julian, Jan, Oliver, Marco, Bahar, Jey Han and all those who had contributed through the online survey. Special thanks to Justin, Rhonda, Julie, Steven and all of the supportive staff of UNIMELB. Thank you to my lab mates Miji, Doris, Nitika, Mohammad, Fei, Siva, Yitong, Qingyu, Diego, Long, Aili, Wei Hao and Wenxi. Thank you to my dearest housemate Alex and my uncles, aunties and cousins in Melbourne. Thank you very much my dearest mum, dad, sis, bro, bro in law, Aidan and Leonard. Not forgetting Zilu, Elly, Alex, friends and supportive colleagues at UNIMAS.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Revision or versioned text documents are texts that have been changed from the original text, where the original source texts are available. Some revisions of documents merely re-phrase or improve writing style, while others can change the meaning of passages (i.e. significant revisions). Revision to documents is commonly practised in many contexts, such as academic writing, legal document preparation, policy refinement, and software requirements review, which generally involve multiple authors. An *edit* is defined as a change that involves operations such as insertion, deletion or substitution of characters or words within a revised text. Authors can make multiple edits for the same text document, and especially in a multi-author environment, multiple edits by different authors can complicate the revision process.

Most of the current collaborative editors are enhancements to text processors, for example Microsoft Word[1] and Overleaf[2], provide the capability to track which author made the change. More advanced versioned document tools that are used for version control such as Apache Subversion[3], not only serve as a repository for versioned documents, but also as an administrative platform to enforce good versioning practises, for instance, standard naming of files and document revision history. In addition, these tools provide the capabilities to link multiple documents together. When a change occurs, other users may also be notified of the change. Despite advancement in these tools, the track changes capability remains limited to highlighting edits at character and word level. In addition to the track changes feature, current word processors have grammar and spell checker features, which also track at word or character level. Hence, users must still manually go through each edit. Furthermore, with the current track changes feature, the authors are still required to read the overall sentences surrounding the edits in order to make sense of the changes, regardless of how small the revision may be. When multiple revisions occur, this task can be overwhelming especially when multiple authors are involved in the writing process. When revising a document within a limited time, some changes can be easily overlooked or unnoticed and the consequences can be more severe if a meaning change goes unnoticed.

[1] https://office.live.com/start/Word.aspx
[2] https://www.overleaf.com
[3] https://subversion.apache.org

Should versioned document tools be able to automatically differentiate the edits between meaning and no meaning change, we hypothesized that this can improve the revision efficiency in terms of attention and time by authors to concentrate on edits with meaning change and may be helpful especially when one draft by an author is passed to another author.

The presentation of the track changes feature in most of the word processors is quite similar where characters or words that have been added or deleted are highlighted. Such presentations are normally generated using the *diff* utility which compares the two versions of the revised text. An example of output using *diff* between an original and revised sentences is provided in Table 1.1. s_o and s_r are syntactically similar, but contain superficially minor differences (see the highlighted words), that nevertheless change the meaning substantially. In this case, the login process is revised to be a compulsory step. These types of sentences are common in revised documents, which makes it challenging to compute meaning change. For example, inserting a word '`not`' is a small syntactic change with a large semantic meaning change. In addition, since edits are widely available, can edits alone be used to assess the impact of revision changes? This research investigates how edits and the words surrounding the edits can support the task of identifying significant revision.

TABLE 1.1: *Diff* between Original and Revised Sentences

Original Sentence, s_o	Revised Sentence, s_r
Surgeon authentication, e.g. user id and password, may be performed for safety and data security reasons	Authentication, e.g. user id and password, is performed for safety and data security reasons
Diff Output	
~~Surgeon authentication~~Authentication, e.g. user id and password, ~~may be~~ is performed for safety and data security reasons	

An automatic identification of significant revisions between two versions of a text document will assist the author to make better informed decisions whether recent changes are of major or minor changes. Assisting authors to assess whether a revision change is meaning change or not can be useful in prioritising revision especially drafting among multiple parties. This can reduce an author's time in reviewing edits especially where the documents can be thousands of pages long and where changes can have profound impact such as public policy documents where changes can have profound impact on how a government mandate is operationalised or in an education environment where editorial changes to student work could inform areas where instructors should focus their teaching.

There are works that automatically classified user edits such as factual and fluency edits (Bronner and Monz, 2012; Daxenberger and Gurevych, 2013), and students' revision behaviour (Zhang and Litman, 2015), while Goyal et al. (2017) look into certain revisions that have higher impact than others. However, none of the work looks into

automatic classification of revisions based on minor and major meaning change or what we defined as significant change identification.

Previous work on revision, whether based on automated or manual analysis, has acknowledged that there are both meaning and non-meaning affecting changes (Faigley and Witte, 1981; Bronner and Monz, 2012; Daxenberger and Gurevych, 2013; Zhang and Litman, 2015; Goyal et al., 2017). Faigley and Witte (1981) proposed a taxonomy to analyse revision according to the meaning change (Figure 1.1). They classified revision into several types. On a general scale, they defined *surface changes* as edits that improved readability without actually changing the meaning of the text, and *text-base changes* as edits that altered the original meaning of the text. These categories were sub-divided. The subcategories for surface changes: *formal change* includes copy editing operations such as correction in spelling, tense, format, etc., while *meaning preserving change* includes re-phrasing. For text-base changes, the sub-categories are *microstructure change* or meaning-altering change which do not affect the original summary of the text and *macro-structure change* or major change which alters the original summary of the text.

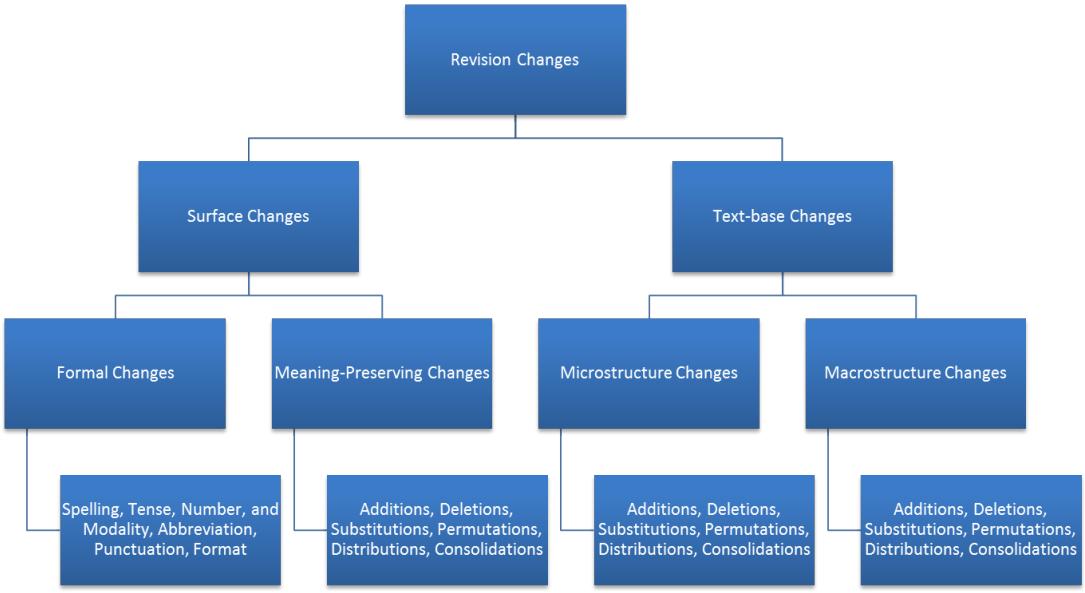


FIGURE 1.1: A taxonomy for analyzing revision (Faigley and Witte, 1981)

Framed by this taxonomy (Figure 1.1), this research investigates revision from a meaning change perspective. We explore how to identify revision that has greater impact or is more significant than another and how to automatically differentiate revision types, in a multi-author revision environment. On the whole, we hope this will improve revision experience especially when transitioning from one draft by one author to another.

1.2 Background of Study

Revision is defined as any change that occurs during the writing process including error corrections, rephrasing and removing or replacing content (Fitzgerald, 1987). Revision can be viewed in two parts (Fitzgerald, 1987):

- the changes made; the by-product of revision (i.e. revised documents)
- the mental workings of revision or in other terms, the processes involved in revision before making direct edits.

Revision, part of the writing process, is a multifaceted process (Faigley and Witte, 1981; Boiarsky, 1984; Hashemi and Schunn, 2014) where the writer is trying to articulate his/her thought. At the same time, the writer is reading and trying to see the text from the reader's perspective, while taking into different considerations like the subject matter, the knowledge of the reader, the style of writing. When the writing flow is not right, the process turns into a troubleshooting process which leads to a problem solving process. Most of all, revision is a recursive process (Boiarsky, 1984; Fitzgerald, 1987). The complexity of the revision process cannot be easily comprehended even for expert writers, let alone for novice writers (Faigley and Witte, 1981; Wallace and Hayes, 1991).

Some collaborative editors include a revision history of all user edits including edit tags to maintain changes, for example Wikipedia[4], creating a large pool of data useful for the purpose of classifying user edits such as factual and fluency edits (Bronner and Monz, 2012; Daxenberger and Gurevych, 2013). Other datasets are not as detailed as Wikipedia, as metadata is limited to revision date and the author who made the revision (Southavilay et al., 2013). Requesting authors to markup each of their revisions such as grammar correction or re-phrase, will disrupt their writing flow, not to mention that it will be time consuming if there is lots of revision. Furthermore, these markup tags might not necessarily be usable for documents of different types.

1.3 Collaborative Writing

Collaborative writing (CW) is defined as two or more authors directly involved in collaborating to produce a written work (Storch, 2005; Ede and Lunsford, 1990; Noël and Robert, 2004). Our research focuses on existing versions of text documents produced in a multi-author environment, thus, this section reviews related works on collaborative writing such as writing strategies and tools to support collaborative writing or computer supported collaborative writing (CSCW) [5]. There are four CW strategies (Noël and Robert, 2004; Scheliga, 2015):

[4] https://en.wikipedia.org/wiki/Help:Page_history

[5] This thesis focuses on collaborative writing (CW), hence, CSCW refers to Computer Supported Collaborative Writing or computer assisted tools for collaborative writing. However, do not be confused with Computer Supported Cooperative Work which Baecker et al. (1995) defined as "computer-assisted coordinated activity carried out by groups of collaborating individuals" that covers a wide range of activities such as communication and problem-solving, including co-authoring a document.

- One author produces a draft and passes it to another author, sequentially;
- Different authors write different parts of a text;
- Only one author writes the text but the text is extended or improved through group discussion;
- Multiple authors write synchronously.

Collaborative writing should not be confused with interactive writing (Button, Johnson, and Furgerson, 1996; Aditomo, Calvo, and Reimann, 2011; Mulligan and Garofalo, 2011; Storch, 2005; Yarrow and Topping, 2001). In interactive writing, an author is given feedback such as an opinion about their writing but the person supplying that feedback is not directly involved in producing the piece of written work. This occurs in teacher feedback and peer review. CW strategies describe the possible ways authors may interact. When we consider the content of a text document revised by multiple authors (Table 1.1), we will see that an automated meaning change detection between revised texts will assist during the transition from one draft by an author to another.

According to a survey conducted by Noël and Robert (2004), they found that despite the existence of specialized collaborative writing tools, most respondents reported still using individual word processors and email as their main tools for writing joint documents. Their findings indicated that users want more than just a tool to write together and recommended functions such as change tracking, version control, and synchronous work for collaborative writing tools. Currently, most CSCW tools incorporate those features and are widely available. Although Wikipedia is not a CSCW tool, CW strategies still apply such as different authors collaboratively writing to contribute various parts of a text. Both Wikipedia and Google Docs are widely used as teaching tools to improve social interaction among writers (Bonk and King, 1995; Hadjerrouit, 2014; Parker and Chao, 2007; Sharples et al., 1993; SchÃűch, 2014; Zhou, Simpson, and Domizi, 2012). In a more recent survey (Scheliga, 2015), a similar finding is obtained by Noël and Robert (2004): writers use a text processor in combination with other digital technologies such as email and content sharing services, instead of using a CSCW tool.

Earlier research in CSCW focuses on supporting collaboration (Fish, Kraut, and Leland, 1988; Haake and Wilson, 1992; Sharples et al., 1993) and designing better user interfaces (Baecker et al., 1993). As technology advances, more research is focused on CSCW tools for the purpose of teaching and learning (Calvo et al., 2011; Parker and Chao, 2007; McWilliams et al., 2013; Hadjerrouit, 2014; Weiss, Urso, and Molli, 2007) including studies on behavioural aspects of CW such as frequency of revisions (Du et al., 2016, visualisation for interaction between authors (Biuk-Aghai, Kelen, and Venkatesan, 2008) and analysis of writing processes for instance, development of ideas during writing (Southavilay et al., 2013) and individual contribution during CW (TREnTIn, 2009).

In summary, CW focuses on the interaction between authors during the writing process. Piolat (1991) stated that it was difficult to conclude with certainty that the use of word processors is always effective in improving writers'revision skills, or that their use necessarily leads to the production of higher quality texts. Even with audit trail data such as which version, which author, what has been edited and the timestamp, the writer lacks input as to whether there has been any substantive meaning change in a revision. Advanced features in CSCW tools have limited support for prioritising revisions and meaning change detection. In the subsequent section, we explore computational works predominantly associated with text revisions.

1.4 Significant Revisions Identification between Revised Text Documents

The section introduces issues related to the task of significant revision identification (SigRevId) between revised text documents which we explore further in this thesis. The question of the significance of revision is particularly challenging in a multi-author environment as different authors might view the impact differently and mainly, how do we determine what actually constitutes a revision with larger impact compared to another or a significant revision for computational implementation. Attempt had been done to rate the importance of the edits according to very important, moderate important, important, neutral and not necessary (Goyal et al., 2017). The edit importance is rated by reviewers, which is used to predict authors' perception of edit importance. However, authors and reviewers might have different perceptions. Furthermore, edits that are more important might not necessarily have higher impact of change and vice versa.

Previous works on revisions whether based on automated or manual analysis have acknowledged that there are both meaning and non-meaning affecting changes (Faigley and Witte, 1981; Bronner and Monz, 2012; Daxenberger and Gurevych, 2013). However, automated classification approach is applied in computational works, while linguistic approach is used in the taxonomy for analyzing revisions (Faigley and Witte, 1981) to categorise revisions to minor and major meaning change. We based significant revisions according to the taxonomy (Faigley and Witte, 1981), thus, the challenge is how do we integrate linguistic approach to a computational method for identifying significant revision.

Revision varies widely with classification of different revisions requiring different annotated data although the texts can be the same. For example, annotated data is prepared for classification of the reason for revisions in students' writing (Zhang and Litman, 2015; Zhang and Litman, 2014). We intend to explore meaning change detection based on linguistic approaches to identify significant revision and a suitable corpus is required for such purpose. The challenge here is to propose an annotation scheme for significant revision identification where authors will agree.

1.5 Aims and Objectives

This research aims to introduce the task of significant revision identification between two versions of a text document in a multi-author environment. We look at versions that come from the same lineage, where one version evolves to another version. For cases where the documents are from different sources, for instance privacy statements from different companies, we do not regard these as versioned documents for the purpose of this research because different companies can derive their own policy independently. However is within our research interest that if the policy is being revised within the same company, the new policy is regarded as a versioned document of the original policy.

This research also aims to develop a computational approach to automatically identify significant changes between versions of a text document, where both versioned documents and original source document are available. The computational algorithm developed in this research applies directly to the end product of revision (i.e. revised text documents) excluding external aspects of text revision such as intention of the revision. Our aim is to create a framework that uses linguistic approach with minimal annotated data as training data. However in order to evaluate the computational approach, a corpus to evaluate the task significant revision identification will be prepared. By having such corpus available, various approaches can be compared to further improve the identification of significant revision.

The aim of computationally identifying significant revision changes is to be able to assist authors in making better decisions in response to the impact of change, especially through prioritising revisions. Edits can be as short as inserting a character. Hence, our focus is on identifying significant revisions for the revised sentences between two versions of a text as sentence-level lets authors comprehend the meaning of the changes better compared to edit (Zhang and Litman 2014).

The research questions explored in this thesis are as follows:

- What are the different kinds of revision changes to be considered as significant revision for revised text documents in a multi-author environment?
- Given two versions of a text document in a multi-author environment, how do we identify significant revisions?
- What are the factors in recognition of textual entailment that can support differentiation of revision changes between revised sentence pairs?

1.6 Research Approach

In this research, as the task of significant revision change identification consists of formalising the various revision types for computational implementation, we use various methods: introspective assessment, user studies, a study of text revisions and various computational approaches. As our aim is to create a computational model that closely

resembles how human evaluate the impact of change, we use the understanding of revision process and linguistic knowledge to derive our conceptual framework. The conceptual framework is validated through user studies with document authors and readers or non-authors. In order to ensure that the computational model is applicable in general, we evaluate on two different document types. As the computational model consists of various components such as recognition of textual entailment systems, we experiment with various approaches to find a suitable approach for our task. The overview of the methodology is presented in Figure 1.2.

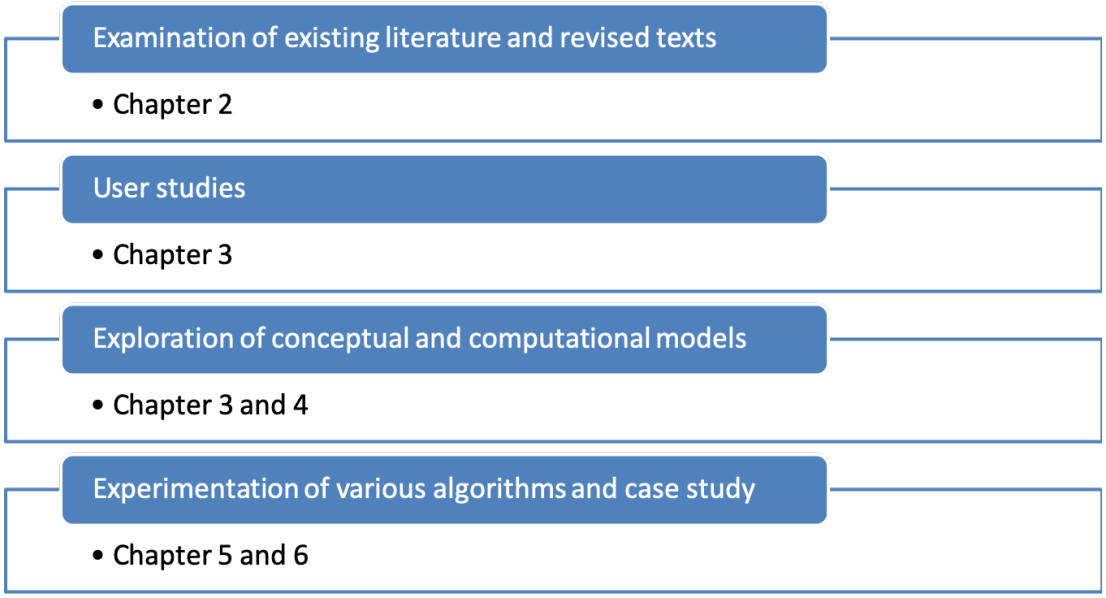


FIGURE 1.2: An Overview of the Research Methodology

1.7 Thesis Overview

In Chapter 2, a review of works related to text revision in a multi-author environment is presented. We will divide this review into two main subsections of revision analysis research: manual revision categorisation and computational revision classification. This chapter will cover the supporting approaches that can assist in SigRevId.

In Chapter 3, we propose a conceptual model for categorising revision changes based on existing work from linguistics and provide a formal definition for the different types of revision changes. Based on feedback from authors and non-authors on revision changes, we discuss what constitute as significant revision changes. We perform introspective analysis of an existing corpus of closely related versioned use case specifications. The introspective analysis, together with the analysis on existing work on psycho-linguistics model, where humans comprehend meaning word by word and follow by phrase by phrase, we highlight that meaning changes can be determined through assessment of both the textual entailment evaluation at sentence level. In addition, the analysis shows properties of this dataset as specific versioned documents.

We derive the different types of revision according to the meaning changes, and provide examples for each revision type. The core of the revision classification is assessment of both the textual entailment directions of the revised sentences. As a result of the revision types classification, significant revision change is defined as major meaning change.

Conceptually, meaning changes can be determined through assessment of both the textual entailment directions of revised sentences as derived in Chapter 3. Based on this conceptual framework, we propose a computational framework to identify significant revision changes between revised documents in Chapter 4. As edits are widely available, we investigate edits in assessing the impact of change before exploring other components such as words surrounding the edits in assessing the impact of change. Firstly we explore the effect of scoping edits at phrase level. Then participating in Semantic Evaluation in the task of Semantic Textual Similarity (Agirre et al., 2016), using a rule based approach, we examine different similarity level of chunks. These elements contribute to the formation of a computational framework for revision classification.

The conceptual framework proposed in Chapter 3 is derived from specific versioned text documents: software requirements specification, the use case specification. In Chapter 5, we derive an annotation scheme to create a corpus purely for evaluation of our computational framework. The corpus we collected consists of academic papers, which is a different type of revision text documents compared to the one we used to derive the conceptual framework. We evaluated the effectiveness of the annotation scheme by measuring the inter-annotator measurements.

Subsequently in Chapter 6, we implement our computational framework using various approaches to recognition of textual entailment and evaluate it on the drafts of academic papers which were annotated earlier. This chapter demonstrates the feasibility of using bi-directional textual entailment evaluation in classifying different types of revisions in addition to edits and other components beyond edits. Based on our results, edit distance approach used in recognition of textual entailment system is suitable for revision types categorisation. We demonstrated that we need to consider more than just edits alone for revision types categorisation such as considerations of dependency trees and sentences entailment. However, edits can support detection of formal changes. Significant revision changes can be detected by evaluating that there is no sentence entailment between the revised sentence pairs.

We conclude our research in the last chapter, considering the broad contribution of the thesis and discuss future work.

Chapter 2

Literature Review

In the previous chapter, the limitations in existing collaborative authoring tools have been identified and a potential approach to improve collaborative authoring systems by enabling authors to automatically accept edits that do not alter the meaning and focus their cognitive attention on edits that do change the meaning of a document is proposed. Even though our aim is a computational framework for identifying significant revision change between revised text documents, our assumption is that implementing a model that resembles natural processes of identifying revisions in texts will be more intuitive and palatable to human users. Thus, human readers can directly understand the detected changes. Hypothetically, the identification of significant revisions will improve the revision experience for authors. In this chapter, related works on text revision are presented. This review is divided into three main parts:

1. Theoretical analysis from manual text revision research to provide linguistic context for our proposed framework,
2. Review of research on measuring edit importance, and
3. Related computational works addressing revision of texts including supporting methods to help us solve our central research problem of automating meaning change detection between versioned text documents.

An overview of this chapter is presented in the Figure 2.1.

(**Note:** Throughout this thesis, the *italic* style is used to identify a new term: *term*, while examples of text revision are presented as: `original sentence` $\longrightarrow$ `revised sentence`.)

2.1 Theoretical Analysis of Text Revision Changes

This section reviews revision research as considered from a non-computational perspective. Text revision can be viewed as the attempt to improve existing written text. It may involve adding or deleting information, or merely re-phrasing so that the message becomes clearer. Our research aim is to create an automatic approach to significant revision change identification between revised text documents in a multi-author environment. However, there are more fundamental questions such as how do we

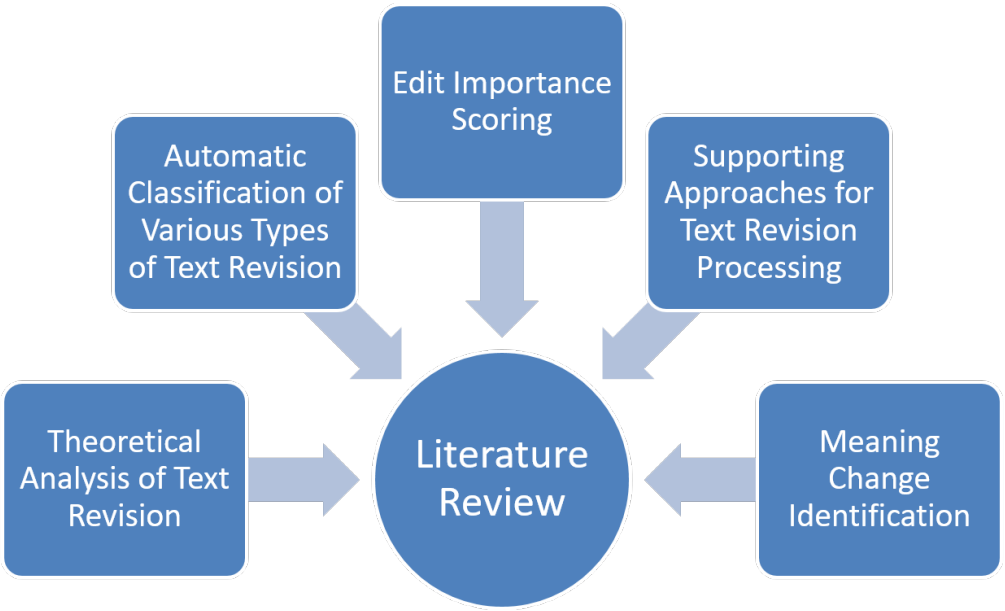


FIGURE 2.1: An overview of Literature Review Chapter

define the significance of a revision? Is there any existing definition for this? Theoretical research on text revision is analysed to aid us in answering these questions and serve as a foundation for analysing meaning change in text revision. To begin, a key study that provides a classification scheme for revision changes, in term of whether the change alters the meaning of the text or not (Faigley and Witte, 1981) is reviewed.

2.1.1 A Taxonomy for Analysing Revision

This section reviews a taxonomy for manual analysis of revision (Faigley and Witte, 1981) (Figure 2.2). This taxonomy differentiates between revisions with no meaning change (i.e. *surface*) and meaning change (i.e. *text-base*) revisions. Faigley and Witte (1981) defined surface change (SC) as revision made with no new information added or old information being removed. This type of change is extended to *formal* and *meaning preserving* changes. Formal change (FC) includes most conventional copy-editing operations. According to the Society of Editors and Proofreaders (Standards director and Ltd, 2016), copy-editing means copying from an existing raw text and checking for consistency and accuracy in preparation for publication. Changes that fall under FC are revisions such as spelling, tense correction, consistent numbering and modality, abbreviation, punctuation, and format.

As defined in the taxonomy (Faigley and Witte, 1981), meaning preserving change (MPC) includes paraphrases of the concepts in the text without altering those concepts. The taxonomy includes revision operations: additions, deletions, substitutions, permutations, distributions and consolidations. They described each of these operations with examples: Addition is defined as "raise to the surface what can be inferred":

`you pay two dollars` $\longrightarrow$ `you pay a two dollar entrance fee.`

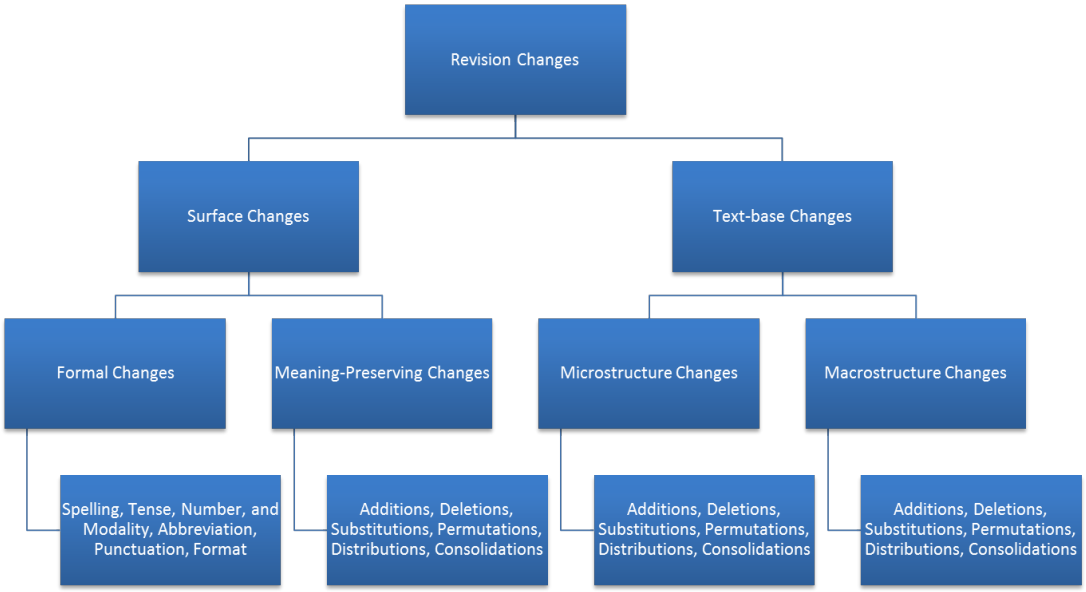


FIGURE 2.2: A taxonomy for analyzing revision (Faigley and Witte, 1981)

Deletion is described as doing the opposite of addition, thus, "a reader is forced to infer what had been explicit":

`several rustic looking restaurants` $\longrightarrow$ `several rustic restaurants.`

Substitution "trades words or longer units that represent the same concept":

`out-of-the-way spots` $\longrightarrow$ `out-of-the-way places.`

Permutation "involves rearrangements or rearrangements with substitutions":

`springtime means to most people` $\longrightarrow$ `springtime, to most people, means.`

Distribution "occurs when material in one text segment is passed into more than one segment or falls into more than one unit":

`I figured after walking so far the least it could do would be to provide a relaxing dinner since I was hungry.` $\longrightarrow$ `I figured the least it owed me was a good meal. All that walking made me hungry.`

Consolidation does the opposite of distribution; "elements in two or more units are consolidated into one unit or can be viewed as exercise to combine sentences":

`And there you find Hamilton's Pool. It has cool green water surrounded by 50-foot cliffs and lush vegetation.` $\longrightarrow$ `And there you find Hamilton's Pool: cool green water surrounded by 50-foot cliffs and lush vegetation.`

If their examples are analysed, addition, deletion and substitution tend to involve words while permutation phrase, distribution and consolidation tend to involve sentences.

When the definitions given in (Faigley and Witte, 1981) are analysed, *no meaning change* is defined as no new information is brought to a text or removing old information, while meaning change is defined as adding of new content or the deletion of existing content in such a way that it cannot be recovered through drawing inference.

Here, whether information is new or old is based on the assessment of texts before and after revision. The notion of information is vague and how is inference deduced and made this computationally feasible? Nevertheless, track changes feature is helpful to present the texts before and after revision for comparison purpose.

Text-base change (TBC) or revision with meaning change is divided to micro-structure change (MiSC) and macro-structure change (MaSC). Faigley and Witte (1981) define major and minor revision changes using Kintsch and Van Dijk's model (Kintsch and Van Dijk, 1978) for comprehending and producing text. In this linguistics and cognitive psychology model, readers are said to comprehend a text phrase-by-phrase and at the same time derive some overall notion of the text called *gist* or *topic* of the text. The meaning of a text is processed at two levels: micro-structure and macro-structure. Micro-structure is all the concepts in the text, both explicit and inferred concepts. Macro-structure characterises the discourse as a whole and represents the "gist" of the text such as a series of labels for section of a text or plot outline. Although Faigley and Witte (1981) stated that macro-structure is a summary of the text, they explained that there was a difference between a summary and macro-structure. Macro-structure can be abstracted from the proposition of a text using series of rules, which we will review in the subsection below.

Faigley and Witte (1981) explained that to distinguish between micro- and macro-structure changes, MiSC would not affect a summary of a text, while MaSC changes the summary. However, MPC falls under the same circumstances as MiSC, that is MPC does not affect the summary. As stated, the difference between MPC and revision with meaning change or TBC is TBC affects the concepts in the text. They did not provide additional explanation on how the concepts were affected.

Faigley and Witte (1981) added more ways to differentiate between micro- and macro-structure changes that is using "constructing summaries for entire texts is to determine if the concepts involved in a particular change affect the reading of other parts of the text." If the entire text is summarised, how do we determine the length or scope of the summary suited for computational implementation. Furthermore, it will not be that obvious which part of the text is affected by the particular change, thus, posing a challenge for computational implementation. Faigley and Witte (1981) stated that micro-structure is all the concepts that can be inferred. This leads to the question of whether there are concepts that do not affect reading of other parts of the text, and if there are, how much of the affected text should be read? Southavilay et al. (2013) have shown that there are a lot of topics overlap when the two revised texts are compared. Hence, when comparing two versions of a text document, the summarisation approach might not be an effective way to determine if the concepts involved in a particular change affect the reading of other parts of the text because the revised texts can be very similar.

To recap no meaning change and meaning change are defined as follow:

No meaning change new information is brought to the text or old information is removed in such a way that it **can** be recovered through drawing inference

Meaning change new information is brought to the text or old information is removed in such a way that it **cannot** be recovered through drawing inference

Faigley and Witte (1981) used Kintsch and Van Dijk's (1978) theoretical model to explain meaning at two levels: a microstructure level is all concepts in a text including those that can be inferred, while macrostructure level represents the "gist" of the text. Based on the theoretical model (Kintsch and Van Dijk, 1978), gist or topic, can be thought of as a series of labels for sections in a text. Macrostructure is essentially the summary of a text and an example of macrostructure is a plot outline. Faigley and Witte (1981) agreed that macrostructure theory is useful but inadequate for distinguishing minor and major revision change. Although Faigley and Witte (1981) considered micro-structure change as minor change of meaning and macro-structure change as major change of meaning, concise definitions are required in order to develop a computational implementation.

The summary of the taxonomy for analyzing revision (Faigley and Witte, 1981) is listed in List 2.1.

LIST 2.1: Summary of taxonomy for analysing revision (Faigley and Witte, 1981)

- There are four types of meaning change in text revision: Formal, Meaning Preserving, Micro-structure and Macro-structure (Figure 2.2).
- *Formal change* has no meaning change and is generally spelling and grammar correction, numbering, copy-editing changes such as formatting. Other than capitalisation, no other exceptional case is listed.
- *Meaning preserving change* is re-wording or re-phrasing or re-arrangement of sentences, including paraphrasing, that does not result in any meaning change. The examples supplied for addition, deletion, substitution and permutation are word- and phrase-level while consolidation and distribution are changes at sentence-level.
- *Micro-structure change* is meaning change which does not affect the summary of the text, which covers all concepts in a text that can be inferred. Micro-structure change is minor revision change.
- *Macro-structure change* is change that affects the summary or the "gist" of the text. Macro-structure change is major meaning change.

The attempt by Faigley and Witte (1981) to classify revision is helpful and this theoretical analysis provides a fundamental understanding of meaning change in text revision. Even though they provided a general definition for the terms in their taxonomy, there is lack of detailed specifications of micro- and macro-structure changes regarding what is considered as "gist" or new information. Faigley and Witte (1981) suggested to use summary approach, however summary can involve summary of a paragraph or summary of the overall text documents. Furthermore, summaries by definition, are precise description which do not contain events or actions but exhibit rather general

or global facts (Van Dijk, 1980). The underlying concept for the taxonomy is "whether new information is brought to the text or whether old information is removed in such a way that it cannot be recovered through drawing inference" (Faigley and Witte, 1981). Our aim is to propose a computational approach that can identify significant revision or revision with higher impact of change. Computationally, what is the approach to draw inference? Thus, an automated system for identification of significant revisions cannot be built on top of this taxonomy directly. Clear definitions of MiSC and MaSC are crucial to ensure that a computationally implementable algorithm can be proposed to identify these revisions. The subsection below attempt to understand micro- and macro-structure in discourse better.

2.1.2 Micro- and Macro-structure in Written Discourse

Based on our earlier review of the taxonomy for analysing revision (Faigley and Witte, 1981), micro- and macro-structure changes lack detailed definitions to enable computational implementation. Faigley and Witte (1981) proposed to use the two-level classification to explain meaning change in revision. This section reviews the existing works that look into the micro- and macro-structure of written discourse.

Faigley and Witte (1981) proposed to use Kintsch and Van Dijk's model (Kintsch and Van Dijk, 1978). According to this theoretical model, a set of propositions ordered by various semantic relations can be used to interpret the surface structure of a discourse. The relations can be either explicit or inferred with additional knowledge such as context-specific and general knowledge. Micro-structure in a discourse is the local structure, for instance, sentences and sequence of sentences that include cohesion, anaphora and inference (Van Dijk, 1980). When deducing meaning, using local sentences and sentence connections alone are insufficient, instead a broader sense or global meaning of the text is required which is the macro-structure (Van Dijk, 1980).

Nonetheless, Van Dijk (1980) proposed general rules that link textual propositions with the macropropositions. These macropropositions are used to define the global topic of a fragment. The rules are considered as semantic derivation or inference rules, where macrostructures are derived from microstructures. These rules are based on the relation of semantic entailment or rather, preserve both truth and meaning. They defined such semantic rules which link text bases, or fragments of these, with macropropositions as *macrorules*. Some of the basic macrorules are:

Deletion or reduction For a sequence of propositions, one or more propositions which are unnecessary to interpret other propositions in the text at the macro-structure level are deleted. The resulting macroproposition is entailed by the microstructural sequence.

Generalisation Propositions can be generalised to a single proposition higher level of abstraction or a global concept. Only the joint sequence of propositions entails the global concept and not each of the propositions in the sequence.

Construction New proposition must be constructed, involving a new predicate to denote the complex event described by the respective propositions of the text. These respective propositions are considered as a joint sequence and is substituted by the new constructed proposition that denotes a global fact of which the micropropositions denote normal components, conditions, or consequences or what is defined as macroproposition. The entailment relation holds between the sequence of proposition the global concept in the knowledge set (or the lexicon), where given the global concept, ideally the necessary propositions in the sequence can be specified.

Even though the macro-structure theory (Kintsch and Van Dijk, 1978) is referred in the taxonomy (Faigley and Witte, 1981), Kintsch and Van Dijk's micro- and macro-structures are based on propositions in a discourse rather than revisions of a discourse. The macrorules of reduction, generalisation and construction given in the theoretical model (Kintsch and Van Dijk, 1978) are too abstract for computational implementation. Micro- and macro-structure for revision changes remain without detailed definition for computational implementation. However, these theoretical understandings serve as the basis to conceptualise micro- and macro-structure revision changes for computational implementation of significant revision identification, which is explained further in the next chapter (Chapter 3).

2.2 Automatic Classification for Various Types of Edits in Text Revision

This section provides a review of approaches to automatic classification of different revision types. An *edit segment* has been defined by Bronner and Monz (2012) as a contiguous sequence of deleted, inserted or equal words by comparing between the original and revised texts. They further defined *fluency edits* as changes to improve on the style and readability and *factual edits* are changes that alter the meaning. They used supervised classification to differentiate the fluency and factual edits in Wikipedia revisions. Daxenberger and Gurevych (2013) proposed to use a predefined 21-edit category taxonomy and used Wikipedia revision histories to perform supervised classification to classify revisions into these categories. Their 21-category taxonomy is divided into three main categories: text-base, surface and Wikipedia policy (vandalism and revert) edits, that is not in the taxonomy for revision analysis (Faigley and Witte, 1981).

Based on the 13-category taxonomy of the semantic intention behind edits in Wikipedia articles, Yang et al. (2017) built a computational classifier of intentions using labelled article edits. This model is used to investigate the effectiveness of edit intention: how different types of edits predict the retention of newcomers and changes in the quality of articles. In a typical collaborative writing, authors do not vandalise their own writing, thus, categories such as vandalism and counter vandalism are not considered.

However, consideration should be given to the other 11 categories in comparison to the four category taxonomy for analysing revision (i.e. formal, meaning preserving, micro- and macro-structure). Furthermore, similar to the reviews of the other supervised classification approaches for revision, we foresee the challenges of implementing their model (Yang et al., 2017) as such a model requires a large corpus of labelled data.

Faigley and Witte (1981) worked on manual revision while Daxenberger and Gurevych (2013) addressed computational analysis revision. Some of the definitions can be linked directly to the taxonomy for analysing revisions (Faigley and Witte, 1981): surface changes correspond to fluency edits while text-base changes correspond to factual edits. Surface changes can also correspond to surface edits which consist of paraphrases, spelling and grammar corrections, relocations and markup edits (Daxenberger and Gurevych, 2013). Other observable similarities between manual and computational revision works are the edit operations: addition, deletion and substitution (Dix, 2006; Faigley and Witte, 1981; Hashemi and Schunn, 2014; Zhang and Litman, 2014; Bronner and Monz, 2012). Other than the edit operations, the edit categories introduced for text-base edits in (Daxenberger and Gurevych, 2013) are not included (Faigley and Witte, 1981). They proposed that text-base edits include sub-categories for templates, references (internal and external links), files and information, each of which is further divided into insertion, deletion and modification types (Daxenberger and Gurevych, 2013).

Using collaborative editors such as Wikipedia and Google Docs not only track user edits, but are also markups for the type of edits made in the document revision history (Bronner and Monz, 2012; Daxenberger and Gurevych, 2013; Southavilay et al., 2013). This information is valuable for automated supervised machine learning, where features are generated and used as training set. The feature sets used are character-level, word-level, part-of-speech, named entities, acronym and language model (Bronner and Monz, 2012). As not all edits can be labelled, Daxenberger and Gurevych (2013) proposed an 'Other' category.

One of the possible edits that fall into 'Other' category is vandalism and reverts. For a free online encyclopedia such as Wikipedia, where most people rely on the information shared, vandalism is a major issue and violating their policies can cause serious problem and thus, this edit category can be considered as a significant change. On the other hand, in a more typical multi-author environment, where it might not necessary be published online or at such scale, where only the authors are allowed to contribute, there might not be any policy intact at all and changes of vandalism is small. The Wikipedia policy edit category (Daxenberger and Gurevych, 2013) cannot be directly applicable to all revisions in a multi-author environment because the policy is to avoid vandalism such as intentionally stated wrong facts while in an atypical multi-author environment, it is a collaborative written work. Although edit categories have been proposed for a typical multi-author environment (Daxenberger and Gurevych, 2013), what is considered as significant revision in this context remains unknown.

Not all versioned text documents have edits well tracked or a revision history available, as most revisions still use a word processor in combination with emails or other sharing services (Scheliga, 2015). We therefore also review works related to computational methods that can assist in classifying text revision when edits and revision history are unavailable. The most relevant work addresses classification of the purpose for revision in augmentative writing (Zhang and Litman, 2015). They proposed a text revision processing pipeline using supervised machine leaning, exploring different features and supervised classification approach to classify the reasons why writers make revisions. Their revision categories consist of two high level categories, i.e. surface and text-base followed by the sub-categories for surface changes which are organization, conventions/grammar/spelling and word usage/clarity, while the sub-categories for text-based changes are claims/ideas, warrant/reasoning/backing, rebuttal/reservation, general content and evidence. The broader categories in Zhang and Litman (2015), text-base and surface changes, correspond to Faigley and Witte's (1981) taxonomy for revision analysis. However, the sub-categories are all different and micro- and macro-structure changes cannot be directly compared to those sub-categories. The sub-categories require annotation in order to be able to differentiate them. Furthermore, they do not consider the impact of revision change.

To summarise, there are works on classification of various types of edits (Bronner and Monz, 2012; Daxenberger and Gurevych, 2013; Zhang and Litman, 2015). However, these works do not look into the meaning change implications of these edits and not all edits have markups. In contrast, we attempt to assist writers in a more meaningful way by presenting the assessment of the significance of the revision. This enables prioritisation of revision changes in multi-author revision. In the remainder of this chapter, we review possible computational components for use in building the framework for automatic identification of significant revision between versioned text documents that we will use in our investigations in later chapters.

2.3 Measuring and Scoring Edits

When an author made an edit, she might view the edit as important, while other co-authors of the same paper might not view that edit as important as the author that made the edit. Edit importance can be subjective depending on the author. When humans are presented with edits, in the scenario where an edit is within a sentence, if the edit is not directly comprehensible (as presented in Figure 2.4) through reading the edits alone, we have a tendency to read the text surrounding the edit. For the case of when a sentence is added or deleted, a reader skims for similar sentence(s), if it exists. These sentences can either be syntactically similar, or have similar or the same meaning. We summarise other cases we need to consider in text revision as below:

- an exact match sentence if there is no change

- sentences with high lexical overlap with minor edits that result in no meaning change, for instance spelling corrections
- sentences with high lexical overlap with minor edits that might change the overall meaning of the sentence
- sentences that have been revised using different words but the meaning remained the same (high semantic similarity with possibility of low lexical overlap), for example paraphrase of a sentence
- sentences that has been revised entirely although there still exists one or two words of overlap.

Here, we review several possible ways to measure edits and score the importance of the edits based on the summary of the revised sentences.

2.3.1 Edit Distance

Edits are changes made to a text. The track changes feature built into word processors, especially in real-time collaboration environments such as Google Docs[1] and Overleaf[2], shows the edits made by authors. Edit distance (ED) is the minimum number of edits (deletion, insertion, or substitution) required to transform one string into another (Navarro, 2001). The underlying calculation of the track changes feature is assumed to be edit distance, similarly to the *diff* approach that focuses on comparing two files to identify the changes made and spelling checkers (Gail et al., 2016). There are a few variance of edit distances such as Levenstein's edit distance (LvD) word error rate (WER), Jaro-Winkler distance and normalised edit distance. LvD and WER will be reviewed in this section while Jaro-Winkler distance and edit distance in general, can be normalised to be used as measurement for string similarity, which will be reviewed in the string similarity section (Section 2.3.2).

2.3.1.1 Levenshtein's Edit Distance

The Levenshtein's edit distance (LvD) (Levenshtein, 1966) between two strings a and b, and the length of a and b is |a| and |b| respectively, is given by $lev_{a,b}(|a|,|b|)$ where

$$lev_{a,b}(i,j) = \begin{cases} max(i,j) & if min(i,j) = 0, \\ min(i,j) \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases} \tag{2.1}$$

[1] https://www.google.com/docs/about/
[2] https://www.overleaf.com

where $1_{(a_i \neq b_j)}$ is the indicator function equal to 0 when $a_i = b_j$ and equal to 1 otherwise, and $lev_{a,b}(i, j)$ is the distance between the first i characters of a and the first j characters of b. The more changes there are between two strings, the higher $lev_{a,b}$.

We provide actual revision sentences as an example to show LvD at word level between a pair of revised sentences, s_o and s_r. If there is no revision at all between s_o and s_r, LvD (s_o, s_r) = 0. If s is revised to t as follows:

s_o = `Surgeon authentication, e.g. user id and password, may be performed for safety and data security reasons.`

s_r = `Authentication, e.g. user id and password, is performed for safety and data security reasons.`,

then LvD(s_o, s_r) = 3, because there are two deletions (`Surgeon` and `be`) and one substitution (`may` $\longrightarrow$ `is`).

2.3.1.2 Word Error Rate

Word error rate (WER) derives from Levenshtein's edit distance and commonly used to evaluate automatic speech recognition systems (Marzal and Vidal, 1993), where there are automatic generated transcription and reference transcript (McCowan et al., 2004). We consider WER because for revised sentence pair, there are original and the revised sentence, which we can consider as generated transcription and automatic transcript. WER is computed as edit distance between a reference word sequence and its automatic transcription, normalised by the length of the reference word sequence (Equation 2.2).

$$WER = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C} \tag{2.2}$$

where

S is the number of substitutions,

D is the number of deletions,

I is the number of insertions,

C is the number of correct words,

N is the number of words in the reference (N=S+D+C)

2.3.2 String Similarity Measurement

String similarity measurement for two strings compares the two strings and quantifies how similar the strings are (Lu et al., 2013). Similarity value of 0 indicates that the two strings are dissimilar while value of 1 indicates both the sentences are the same, while similarity value closer to 0 shows less similarity while closer to 1 shows the two strings are more similar. We consider similarity approaches that utilise edit distances because previous works on text revision works focus on edits (Bronner and Monz, 2012; Daxenberger and Gurevych, 2013; Goyal et al., 2017; Zhang and Litman, 2015). Here, we review Jaro-Winkler similarity and normalised edit distance which can also be used for alignment of various revised sentences.

2.3.2.1 Jaro-Winkler Similarity

Jaro-Winkler distance is another variants of edit distance. In order to measure the similarity of revised sentences, we consider Jaro-Winkler (Winkler, 1990) string metric. Jaro-Winkler algorithm is a modification of Jaro algorithm (Jaro, 1989). Both the equations are computed as below:

$$sim_j = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) & \text{otherwise} \end{cases} \tag{2.3}$$

Where:
$|s_i|$ is the length of the string s_i;
m is the number of matching characters (see below);
t is half the number of transpositions (see below)

$$sim_w = sim_j + \ell p(1 - sim_j), \tag{2.4}$$

where:
sim_j is the Jaro similarity for strings
$s_1 and s_2$
ℓ is the length of common prefix at the start of the string up to a maximum of four characters

p is a constant scaling factor for how much the score is adjusted upwards for having common prefixes.

p should not exceed 0.25, otherwise the distance can become larger than 1. The standard value for this constant in Winkler's work is p=0.1.

2.3.2.2 Normalised Edit Distance

Levenstein's edit distance (Section 2.3.1.1) values are normalised (Equation 2.5) to [0, 1] (Attig and Perner, 2011) which is used as a string similarity measurement (Navarro, 2001). Conceptually, when these values are applied to string similarity, the value of 1 indicate complete lexical overlap, while value of 0 indicates no minimal overlap, likewise value closer to 0, less lexical overlap and closer to 1, higher lexical overlap. When applied to revised sentences, revised sentences with high lexical overlap but with minor edits will likely to have high string similarity values.

$$1 - \frac{editdistance}{lengthofthelargerofthetwostrings} \tag{2.5}$$

Nevertheless, edit distance based approaches only indicate surface changes and cannot measure meaning change. This clearly shows the limitation of current change detection features in supporting meaning change detection. Conceptualising edit importance will be tricky because if edit importance is measured according to word overlap, edit distance is a good indicator. However if edit importance is based on meaning change, edit distance alone might not be that helpful.

2.3.3 Sentence Similarity Measurement

Sentence similarity measurement between two natural language sentences quantifies how semantically equivalent the two sentences are (Achananuparp, Hu, and Shen, 2008). Therefore, semantic similarity measurement that looks into linguistic properties, such as semantic relations (Achananuparp, Hu, and Shen, 2008). Sentence similarity scores are similar to string similarity values, where scores nearer to 0 indicate less similarity while scores closer to 1 indicate more similarity.

The previous section has provided reviews of string similarity measurements where no additional semantic component is added but here, we consider similarity measurements which use different linguistic properties such as part-of-speech, WordNet, an electronic lexical database (Miller, 2009) and word order (section 2.6.1 reviews paraphrase approaches to support meaning preserving change detection). There are existing works which consider one or all of those (Fernando and Stevenson, 2008; Lee, Chang, and Hsieh, 2014; Li et al., 2006; Mihalcea, Corley, and Strapparava, 2006; Vo, Magnolini, and Popescu, 2015). These works have been shown to be promising for sentences that have been paraphrased or re-phrased with the same meaning, which is essentially just a type of revised sentences (the review on approaches to identify paraphrases is provided in section 2.6.1). For the case of revision, we are concerned with more than just sentences that have been paraphrased.

2.3.4 Pearson Correlation Coefficient

One way to score edits is to observe the correlation between various measurements of edits such as similarity measurements (Section 2.3.2 and 2.3.3) human annotation similar to the work done by Goyal et al. (2017). Correlation is numerically measured using correlation coefficient and a widely used correlation coefficient is Pearson correlation coefficient, r (Benesty et al., 2009). The correlation coefficient calculates the strength of the relationship between two variables (i.e. the measurement of edits and human annotation on edit importance). The values range between -1 and +1, where -1 is a perfect opposite correlation, 0 means no relationship between the variables and +1 means a perfect correlation. If r value closer to 1, the measurement correlates better with the significance, while opposite correlation is ob- served for negative r value. When r value is closer to 0, weak correlation between the variables.

2.3.5 Scoring of Edit Importance

Goyal et al. (2017) proposed to use various features for scoring the edit importance to predict authors' perception of edit importance. They manually added, modified and deleted information in news corpus to create factual edits while making changes to writing style or paraphrasing such as synonymous words/phrases/number and

changing from active to passive voice to create new versions. They employed reviewers (or non-authors) from Amazon Mechanical Turk[3] to rate the edit importance of each change as very important, moderately important, important, neutral and not necessary for review. They proposed to use factual and fluency edits but to sub-divide factual edit into information modify, information delete and information insert, while fluency edit is sub-divided to lexical paraphrase and transformational paraphrase. Lexical paraphrase is changes to the textual using synonymous words/phrases/numbers. Transformational paraphrase is changes to the sentence structure such as from active to passive voice. They then extracted features related to change or relevance for supervised modelling of edit importance. Change-related features correspond to factual edits and relevance-related features correspond to fluency edits. Change-related features are scored using a heuristic approach; factual edits are weighted higher than fluency edits, and revised sentences with higher count of differences for the PoS and named entities including dependency changes, edit counts and readability will also have higher weights. Relevance-related features, on the other hand, are scored according to the relevance of the sentences to the overall text and the position of the sentence in the text. They demonstrated that these scores correlate to reviewers' annotation of edit importance, with features of PoS tags and change in dependency tuples having the highest correlation.

Although Goyal et al. (2017) demonstrated that their scores correlated to human annotation (i.e. Benesty et al., 2009), there was no attempt to directly breakdown the analysis by revision type, for example according to minor and major meaning changes as presented in the taxonomy for analysing revision (Faigley and Witte, 1981). The revised sentences used in (Goyal et al., 2017) are manually revised and rated by Turkers according to their ratings system of edit importance. There are various types of revised sentences where there are circumstances where we might need to consider both string and semantic similarity. For instance, minimal lexical overlap with the same meaning can be considered as meaning preserving change, which is essentially just one of the four categories of revision changes. We assume revised sentences that have been re-phrased will have high semantic similarity values but low string similarity. For revised sentences with spelling correction, we assume that the sentences will very likely have both high values for string and semantic similarities. Hence, edit importance depends on the type of revisions made to the sentences. Thus, edit importance can be viewed as changes to both lexical and meaning of a sentence. Due to the variability in revisions, observation is required to determine suitable similarity measurement and threshold for the types of revision.

[3]https://www.mturk.com/

2.4 Text Revision Processing

In this section, related computational works to process revised texts are reviewed. Zhang and Litman (2015) proposed a pipeline for supervised classification of text revision. They highlight three main processes as listed below:

revision extraction process of identifying and extracting changes in revised texts,

revision categorisation human annotation of different types of revisions, and

revision classification process to differentiation the types of revisions.

Revision extraction and revision classification are generally automated efforts. Revision categorisation on the other hand, focuses on how revision are categorised based on human feedback. Here, we review possible approaches for revision extraction such as summarisation and visualisation in collaborative writing, the *diff* utility, and sentence alignment between versioned text documents.

2.4.1 Summarisation and Visualisation in Collaborative Writing

Hashemi and Schunn (2014) presented a tool to assist in peer review learning approach by summarising changes such as the number of edits between drafts before and after peer review. They first split the original documents into sentences and then built on the output of Compare Suite [4] to count and highlight changes in different colours. This is used to help professors summarize students'changes across papers before and after peer review. Although the writing environment in this work is based on peer reviews rather than multiple authors working on writing the same piece of text, providing authors with general summary of the revisions made, can provide an overview of how extensively the text has been revised.

Southavilay et al. (2013) proposed visualization approaches for analysing writing processes such as writers' interaction and shift of topics in a collaborative writing environment. Their effort utilised the data generated through Google Docs [5], an online collaborative writing tool. Three proposed visualisation approaches are:

- A revision map which summarises what has been edited at paragraph level all through the course of writing
- A topic evolution chart which uses a probabilistic topic model to extract the topics and presents how topics evolve during the writing process
- A topic-based collaboration network, which present the topics in relation to the authors' contributions and collaboration.

Our focus is on the topic evolution chart (Figure 2.3) to observe the possibility of topic extracted being used as comparison for revision changes. Southavilay et al.

[4] https://comparesuite.com/
[5] https://www.google.com/docs/about/

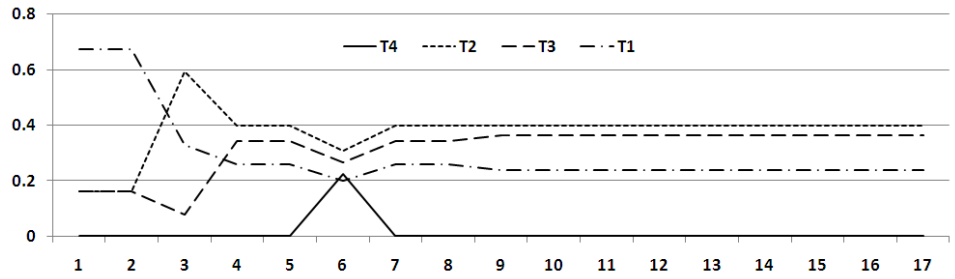


FIGURE 2.3: Topic Evolution Chart of four topics (T1, T2, T3 and T4) for 17 versions according to the percentage the topic is covered in a version (Southavilay et al., 2013)

(2013) define *topic* as a cluster of words that frequently appear together in a version and each version consists of set of topics. The topics are extracted using DiffLDA, which is a combination of the GNU *diff* utility (review in Section 2.4.2) and Latent Dirichlet Allocation (LDA), a probabilistic topic modelling method. They provided an example of a topic evolution chart where a document had been revised 17 times with four topics extracted (Figure 2.3). The topics in the topic evolution chart are presented as percentages covered by the topic within that version, for instance, version 1, T1 covered about 66%, while T2 and T3 captured about 17% each respectively. They did not state how the percentages are generated for the topics.

Initially, we viewed the idea of topic extraction as a possible computational way to extract macro-structure. Van Dijk (1980) explained that macro-structure represented the "gist" of the text such as a series of labels for section of a text or plot outline (reviewed in Section 2.1). When we evaluate the approach in (Southavilay et al., 2013), we observe that the topics are consistent, especially between drafts or versions that are revised right after another (Figure 2.3). Assume that each version is revised by a different author because as for our research, we attempt to identify significant revisions between drafts by different authors. In the case of version 9 to 17, there is a lack of substantial change in topic distribution, thus, if authors continued to revise by version 9, there was no meaning change? Moreover, the topic words provided no indication of which parts of the document had changed between revisions, hence, when one author passed it to another, the other author still needed to read all changes to understand the impact of the previous revisions step. Furthermore, they had also shown that versioned texts have a lot of similar words which could influence the similarity measurement. Nevertheless, the approach proposed by Southavilay et al. (2013) to extract topics at paragraph level can be considered.

The dataset used by Southavilay et al. (2013) are drafts produced by authors in a more typical CW environment compared to most computational works in revision which uses Wikipedia dataset (Bronner and Monz, 2012; Daxenberger and Gurevych, 2013), a public dataset. They demonstrated that working on drafts produced by authors at smaller scale was feasible.

2.4.2 *Diff* Utility

This section reviews a commonly used tools for finding the difference between two files. The *diff* utility (Free Software Foundation, 2016; Neuwirth et al., 1992; Wang, DeWitt, and Cai, 2003; MacKenzie, Eggert, and Stallman, 2003). *Diff* has been used to present the output of the differences between two revised text documents (Neuwirth et al., 1992). In text revision research, *diff* is used as a pre-processing step to extract out *edits* or words that had been added, deleted or modified on a text document (Bronner and Monz, 2012; Southavilay et al., 2013). The algorithm behind the *diff* utility is typically a longest common subsequence (Myers, 1986) or Levenshtein distance algorithm (Levenshtein, 1966) or more generally known as edit distance, which processes files line-by-line to identify insertions and deletions. An explanation on Levenshtein distance was provided in Section 2.3.3.

In order to show how *diff* works, we compare two actual revisions of a text document, each version by different authors using a version of *diff*, LaTeX *Diff* [6] as shown in Figure 2.4. The output of LaTeX *Diff* in Figure 2.4 is similar to the output of a track changes feature in most text processors. As demonstrated in the example, although the sentences in the two versions of the text document align quite well and *diff* output shows readers which words had been modified, the output has no indication whether there is any meaning change. For instance, in the example (Figure 2.4), `August 2010` $\longrightarrow$ st. There was no indication of meaning change unless a reader read "21st `and` 22nd `August 2010`". Also in this particular instance, even though reader is notified of a deletion: "`August 2010, and is`" and an addition: "`August 2010. The 622 192 messages are`", a typical human reader would observe that there is a deletion of "`, and is`" and an addition of "`The 622 192 messages are`". The *diff* utility cannot scope the edits accordingly because *diff* does not contain any information about sentence structure. We foresee an improved scoping of the edits lets readers make better judgement of meaning changes.

Nevertheless, *diff* is able to detect and serve as an indicator to the reader that a word or sentence has been edited. In addition, the sentences between the two versions of the text document are aligned well. This supports *diff* utility as a pre-processing step between two versions of a text document to align the sentences and to extract edits, but *diff* utility cannot be used solely for meaning change detection. For meaning change detection, we require processing beyond edit extraction.

2.4.3 Sentence Alignment

There is existing research that looks into aligning sentences in revised texts to detect if a sentence has been re-written between the first and last drafts of a student's essay in an interactive writing context, (Zhang and Litman, 2014). Thus, given two revised texts, processing at sentence level is a reasonable starting point. This then requires

[6] https://3142.nl/latex-diff/

The first new dataset is TWITTER, and consists of ~~622192 messages from Twitter. This data was~~ 5000 messages randomly selected from a larger body of 622192 messages collected from the Twitter Streaming API over a single 24-hour period between 21~~August 2010~~ st and 22~~August 2010, and is~~ nd August 2010. The 622192 messages are a 1% representative sample of the total public status updates on that day. ~~From this collection, we randomly selected 5000 messages . Each message~~ Each of the 500 selected messages was then annotated by speakers of three languages, English, Japanese and Mandarin Chinese. In total, 2 trilingual annotators and

FIGURE 2.4: LaTeX *Diff* Output

alignment of revised sentences. First, Zhang and Litman (2014) had a human annotator aligns the sentences. Adapting from sentence alignment work for monolingual corpora (Nelken and Shieber, 2006), the aligned sentences were used as training for a logistic regression classifier. This was followed by manually differentiation of different types of aligned sentences: no change or keep, modify, delete or add. Lastly, if the sentence from the original text was aligned to more than one sentence or if more than one sentence from the original text was aligned to one sentence in the revised text, the aligned sentences were labelled according to the edit operations. This work focuses on the first and last drafts of the essay by the same author. When the approach is applied to versions of texts by different authors, manually aligning sentences can become complicated because as demonstrated in the topic evolution chart (Figure 2.3) in (Southavilay et al., 2013), the original topic by one author might not even exist throughout the version and new topics could be introduced at any version by authors. Despite that, no analysis of the significance of revision changes was made. However, Zhang and Litman (2014) provided important insights into processing revised texts: sentence order is important for sentence alignment, alignment might not necessarily be a one to one alignment; and sentence alignment is a required process to detect if the sentence has been re-written.

Basically what we require is a good enough similarity measure to support alignment of revised sentences before significant revision evaluation. Most definitions of string similarity are application or approach specific (Lin, 1998; Rieck and Wressnegger, 2016) including which string metric to be used has also been shown to be application specific (Cheatham and Hitzler, 2013). Most work on sentence similarity measures are evaluated on paraphrase corpora (Cohen, Ravikumar, and Fienberg, 2003; Achananuparp, Hu, and Shen, 2008; Li et al., 2006; Lee, Chang, and Hsieh, 2014; Mihalcea, Corley, and Strapparava, 2006; Fernando and Stevenson, 2008). Although here we focus on sentence similarity measures for the task of natural language processing,

there are works on similarity measures applied to various tasks such as outlier detection (Boriah, Chandola, and Kumar, 2008), information retrieval (Metzler, Dumais, and Meek, 2007), and ontology alignment (Cheatham and Hitzler, 2013).

Given two monolingual texts or texts with the same language, usually, the aim of sentence alignment is to find two sentences or texts that convey the same information (Barzilay and McKeown, 2001; Nelken and Shieber, 2006; Sanchez-Perez, Sidorov, and Gelbukh, 2014; Liu et al., 2014). There are existing sentence alignment methods applied to monolingual corpora for various applications such as e-commerce policy statements (Liu et al., 2014), encyclopedia entries (Nelken and Shieber, 2006), parallel texts in statistical machine translation (Wołk and Marasek, 2014; Xu, Max, and Yvon, 2015), and text summarisation (Hirao et al., 2004). We identify approaches to sentence alignment developed for monolingual corpora as a possible strategy for aligning revised sentences. Although we consider approaches used in sentence alignment between monolingual texts, we are aware that for revised texts there can be cases involving minimal lexical variability unlike monolingual texts, where the authors are different with their respective writing styles. As a result, when we compare sequential revisions of a text, they may be very similar with only slight changes at sentence level, that in turn change the overall meaning of the sentence (see example in Figure 2.4).

Barzilay and Elhadad (2003) opted for a machine learning approach to align sentences between comparable monolingual texts that convey the same information that have little surface resemblance or less lexical overlap. First the paragraphs from the monolingual texts were clustered into groups. Then, manually aligned text pairs were used to train a binary classifier, and used to predict whether two sentences should be aligned or not. They added another process to measure the similarity of the sentences predicted by the classifier. Their idea was to first consider alignment at global level before evaluating the similarity of sentences at the local level. When we consider revision at sentence level, edits can either be edited words within the revised sentence or a full sentence that has been added or deleted, hence in such cases, there might be no direct sentence to be aligned. This work suggests that both global and local alignments using similarity measurement can be useful for extracting revised sentences.

Paraphrase is considered as a meaning preserving change (MPC) (Section 2.1.1), while MPC is one of the four categories of meaning change for text revisions (i.e. formal, meaning preserving, micro- and macro-structure changes). Although we consider sentence alignment methods developed for monolingual corpora for the purpose of aligning sentences that convey the same meaning between revisions, our aim is broader: our purpose is to identify sentences that are related based on different types of meaning change while alignment in monolingual corpora always targets sentences with the same meaning. In the remainder of this section we concentrate on works addressing sentence alignment between revised texts.

In brief, approaches to sentence alignment for monolingual corpora usually are designed for the purpose of searching for texts that convey the same information, where

the underlying approach is sentence similarity (Barzilay and McKeown, 2001; Nelken and Shieber, 2006; Sanchez-Perez, Sidorov, and Gelbukh, 2014). Zhang and Litman (2014) have demonstrated that the sentence alignment for monolingual corpora can be adapted to the context of detecting whether a sentence has been re-written between the first and last drafts of an essay by an author. However, none of these researches address the significance of revision changes and most of the proposed methods require manually aligned sentences. We do not require high accuracy for sentence alignment, rather, a reasonably good alignment of revised sentence pairs is sufficient before proceeding to significant revision evaluation. Therefore, instead of requiring manual alignment of sentences, we consider sentence similarity to be sufficient to support alignment to meet our broader aim of significance revision changes identification.

As a summary for sentence alignment, revision with formal change (i.e. grammar or spelling mistakes) can have high lexical overlap, while revision with meaning preserving change may have high semantic similarity with low lexical overlap. There is also the possibility that revised sentences that have high lexical overlap with minor edits which change the meaning entirely. Sentence similarity measure is not sufficient for producing the significance of revision change. However, it is a useful starting point for aligning revised sentences for further processing.

2.5 Evaluation of Text Revision Classification

There is no standard corpus for text revision processing. Furthermore, different research addresses various types of revisions (Bronner and Monz, 2012; Daxenberger and Gurevych, 2013; Zhang and Litman, 2014; Goyal et al., 2017). Nevertheless, the annotated data used in these works are rated by humans. As our effort focuses on minor and major meaning changes, for human categorisation effort of revisions, we review inter-rater reliability measurements (Subsection 2.5.1).

2.5.1 Inter-rater Reliability Measurement

Generally, inter-rater reliability is used to measure whether the raters agree with each other according to a rating scheme, where higher reliability means agreement on the rating scheme (Gwet, 2014). In addition, our aim is to measure the agreement between the raters when judging meaning change in revision.

Common inter-rater reliability measurements include simple agreement in percentage (%) (Formula 2.6), Scott's π (Formula 2.8) and Cohen's κ (Formula 2.7) (Pustejovsky and Stubbs, 2012). π has been shown to be more reliable than κ if there is only two categories (Limited, 2016). Therefore π is used to measure two raters two categories while, κ is used to measure two raters four categories. For cases where there are more than two raters or non authors, Fleiss' κ (Formula 2.9) is used (Pustejovsky and Stubbs, 2012).

Simple agreement

$$(\%) = \frac{A}{N} * 100 \tag{2.6}$$

where,

A is the number of ratings rated the same, and

N is total number of ratings

Cohen's kappa,

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{2.7}$$

where,

p_o is the relative observed agreement among raters, and

p_e is the hypothetical probability of chance agreement.

Scott's pi

$$\pi = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \tag{2.8}$$

where,

$Pr(a)$ is calculated observed agreement, and

$Pr(e)$ is calculated using joint proportions

Fleiss' kappa,

$$\kappa = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e} \tag{2.9}$$

where,

$\bar{P} - \bar{P}_e$ gives the degree of agreement actually achieved above chance, and

$1 - \bar{P}_e$ gives the degree of agreement that is attainable above chance.

Krippendorff's alpha, α (Formula 2.10) is a inter-reliability measurement which considers disagreement between raters (Krippendorff, 2011) with four different types of calculation: Nominal, Interval, Ordinal and Ratio. α value closer to 1 indicates perfect reliability while $\alpha = 0$ indicates the absence of reliability (Krippendorff, 2011). Nominal type treats each of the categories as singular category, interval type treats the categories as quantitative values, while ordinal type treats the categories in an order form and ratio type treats each of the categories as a ratio to another. In the case of revision categories (i.e. formal, meaning preserving, micro- and macro-structure changes), the nature is nominal. However these categories can be viewed as ordinal where formal revision has the least impact of change, gradually increasing to meaning preserving change, follow by micro-structure revision, with macro-structure revision as the highest impact of change. Thus, for α values, the reliability are calculated based on two types of measurement: $\alpha_{nominal}$ and $\alpha_{ordinal}$.

Krippendorff's alpha,

$$\alpha = 1 - \frac{D_o}{D_e} \tag{2.10}$$

where,

D_o is the disagreement observed, and

D_e is the disagreement expected by chance.

2.5.2 Evaluation Measurements for Revision Classification

Section 2.2 reviews automated classification of various revision types which is generally defined as a classification task. Hence, this section reviews evaluation measurements for automated classification of various revision types. Automated classification of revisions involves labelling revisions in term of specific categories (Bronner and Monz, 2012; Daxenberger and Gurevych, 2013; Zhang and Litman, 2015). Based on the taxonomy for analysing revisions (Faigley and Witte, 1981), our task is classifying significant change. Revisions are classified into one of the four categories (i.e. formal, meaning preserving, minor and major meaning change), and is therefore a multi-class classification problem (Aly, 2005). In the case of revision types categorisation in this research, C_i is category for i, one of the four categories. Further definitions for true positive (tp), true negative (tn), false positive (fp) and false negative (fn) (Lever, Krzywinski, and Altman, 2016) according to revision type classification are provided as below :

tp_i The approach assigned the label C_i to versioned sentence pair, the same as the annotator.

tn_i The approach and the human annotator agreed the versioned sentence pair does not represent a matched revision pair.

fp_i The approach produces as the label C_i for the versioned sentence pair while annotator indicated as not that category.

fn_i is The approach produces a different label while the annotator annotated as C_i for that versioned sentence pair.

The evaluation measures are adopted from Sokolova and Lapalme (2009) and are summarised in the Table 2.1. Recall, precision and F-score are three more commonly used evaluation measurements in classification tasks (Van Asch, 2013). In the case of significant revision identification, precision for a revision type is the fraction of correctly identified revision types for the revised sentence pairs labelled as that revision type, while recall for a revision type is the fraction of correctly identified revision type for the total amount of revised sentence pairs for that revision type as annotated by human annotators. F_1-score is the harmonic average between precision and recall. Significant revision identification involves categorising four classes and by averaging the results either using macro- or micro-average, providing a general view of the overall results. Macro-average is the average based on equal weight, while micro-average averages according to each revision type. The definitions are presented in Table 2.1.

TABLE 2.1: Evaluation Measures with μ as micro-averaging and M as macro-averaging (Sokolova and Lapalme, 2009)

Measure	Formula	Evaluation Focus
$Precision_\mu$	$\frac{\Sigma_i^l tp_i}{\Sigma_i^l (tp_i + fp_i)}$	Agreement of the data class labels with those of a classifier if calculated from sums of per-revision decisions
$Recall_\mu$	$\frac{\Sigma_i^l tp_i}{\Sigma_i^l (tp_i + fn_i)}$	Effectiveness of a classifier to identify class labels if calculated from sums of per-revision sentence pair decisions
$Fscore_\mu$	$\frac{(\beta^2+1) * Precision_\mu * Recall_\mu}{\beta^2 Precision_\mu + Recall_\mu}$	Relations between data with positive labels and those given by a classifier based on sums of per-revised sentence pair decisions
$Precision_M$	$\frac{\Sigma_i^l \frac{tp_i}{tp_i + fp_i}}{l}$	An average per-class agreement of the data class labels with those of a classifier
$Recall_M$	$\frac{\Sigma_i^l \frac{tp_i}{tp_i + fn_i}}{l}$	An average per-class effectiveness of a classifier to identify class labels
$Fscore_M$	$\frac{(\beta^2+1) * Precision_M * Recall_M}{\beta^2 Precision_M + Recall_M}$	Relations between data with positive labels and those given by a classifier based on a per-class average

2.6 Meaning Change Identification

In Section 2.1, the taxonomy for analysing revision (Faigley and Witte, 1981) is reviewed, where surface change (SC) is revision without meaning change with two subcategories: formal change (FC) and meaning preserving change (MPC), which include paraphrase, while text-base change (TBC) is change that alters the meaning. This section reviews research that addresses meaning change detection.

2.6.1 Paraphrase Recognition

Paraphrase is the re-wording of sentences or phrases without changing the meaning (Bhagat and Hovy, 2013; Boonthum, 2004; Zhao and Wang, 2010). Both paraphrase and SC have no meaning change while revised sentences that are not paraphrases

of one another, very likely fall into TBC. In this section, we review related works to paraphrase in relation to SC and TBC.

In *paraphrase identification*, given two texts, we detect if the texts are paraphrases of each other. A high semantic similarity value does not always mean that the two texts are a paraphrase of each other. For instance, a small spelling correction produces high semantic similarity but is not a case of paraphrase. For example:

s_o = `The size of the corpus is important in methods based on distributional similarity.`

s_r = `The size of the corpus is important due to the use of statistical measures in most of the proposed compositionality detection methods.`

Paraphrase identification can be considered as an evaluation approach after the revised sentences have been aligned. Considering at lexical, phrase, sentence and discourse level, paraphrase recognition/identification is the task of identifying the following patterns (Bhagat and Hovy, 2013):

- Substitution such as synonym, antonym, converse, Actor/Action, pronoun/co-referent, Verb/"Semantic-role noun", Manipulator/Device, General/Specific, Metaphor, Part/Whole, Verb-preposition/Noun, External knowledge
- Changes such as voice, person, tense, aspect
- Repetition/Ellipsis
- Function word variations
- Conversion such as Verb/Noun, Verb/Adverb
- Semantic implication

Paraphrase is applicable to various types of natural language tasks (Zhao and Wang, 2010), information retrieval (Wallis, 1993; Zhang et al., 2015), question answering (Berant et al., 2013; Boonthum, 2004; McKeown, 1979), information extraction (Barzilay and McKeown, 2001; Shinyama and Sekine, 2003; Regneri, Wang, and Pinkal, 2014), text summarisation (Patil, Bewoor, and Patil, 2014) and automatic evaluation of machine translation (Barzilay and McKeown, 2001; Jurafsky and Martin, 2014; Liu, Dahlmeier, and Ng, 2010; Madnani, Tetreault, and Chodorow, 2012). The approaches used in paraphrase identification are quite similar to sentence similarity measures approaches ranging from statistical model, both supervised (Pham et al., 2013; Liu, Dahlmeier, and Ng, 2010) and unsupervised (Barzilay and Lee, 2003), semantic similarity measures (Fernando and Stevenson, 2008; Barzilay and McKeown, 2001; Wang and Callison-Burch, 2011) and combination of lexical and syntactic information (Lee, Chang, and Hsieh, 2014; Zhang et al., 2014), bi-directional textual entailment (Androutsopoulos and Malakasiotis, 2010; Μαλακασιώτης, 2011; Romano et al., 2006; Giampiccolo et al., 2007; Watanabe et al., 2013), even machine translation evaluation metrics (Vo, Magnolini, and Popescu, 2015; Madnani, Tetreault, and Chodorow, 2012).

Although there are a considerable number of approaches we can consider for the task of paraphrase detection, notably, significant revision detection can go beyond paraphrase detection. One approach is bi-directional textual entailment for paraphrase detection. With regard to micro- and macro-structure changes in the earlier section (Section 2.1.2), Van Dijk (1980) introduces entailment by stating that there is a relationship between the meaning of the text and the topic. Meaning at macro-structure level entails the topic and meaning at the micro-structure level entails the meaning at the macro-structure level. If meaning changes at micro- or macro-structure level, the meaning might not entail at the different level. We consider extending the use of textual entailment to infer the meaning changes of texts. We will elaborate on this aspect in the subsequent section on recognition of textual entailment (RTE).

2.6.2 Recognition of Textual Entailment

By definition, a text entails when we can infer a text from reading another text, or when a human reader reads the first text to be true, the second text is most likely to be true. As stated for the taxonomy for analysing revision (Faigley and Witte, 1981), in order to determine meaning preserving changes consider what can be inferred explicitly or forced to infer what had once been explicit from the revised sentence. If this holds, what cannot be inferred will lead to meaning changes. *Recognising textual entailment* (RTE) is the task of identifying whether a piece of text can be plausibly inferred from another (Dagan and Glickman, 2004; Dagan et al., 2013; Sammons, Vydiswaran, and Roth, 2011; Tatar et al., 2009). Hence, we focus on recognition of textual entailment (RTE) to identify the meaning change in revisions. Furthermore, RTE approaches considered many aspects which correlate to edit importance (refer to Table 3.8).

Recognition of textual entailment (RTE) does not only produce the entailment outcome of the two sentences, the entailment outcome is dependent on the directional relation of the sentences. This gives an advantage of RTE over other meaning change methods to support significant revision identification as this is inline with our proposed conceptual framework to evaluate the entailment outcome according to the directional relation of the revised sentence pairs. In addition, RTE can include aspects of NLP such as lexical meaning, syntactic information and directional relation (Tatar et al., 2009), hypothetically, able to assist in meaning changes detection better. RTE has been applied to various natural language tasks (Glickman, 2006; Ghuge and Bhattacharya, 2014; Dagan, Glickman, and Magnini, 2006; Dagan et al., 2013) such as question answering (MacCartney et al., 2006; Pakray, 2011), information retrieval (Clinchant, Goutte, and Gaussier, 2006), information extraction (IE) (Shnarch, 2008; Tatar et al., 2009) and question answer (QA) (Dzikovska, Nielsen, and Leacock, 2016; MacCartney et al., 2006; Pakray, 2011), even though none looks into revision analysis.

While we consider paraphrase recognition to evaluate whether revised sentences are re-phrased of one another, here we consider RTE to evaluate whether the truth still holds between revised sentences. Bos (2014) define RTE as the task to decide if a text contains new information with respect to the other text. Thus, given two texts (Text, T

and Hypothesis Text, H), T is said to entail H (denoted as T $\Rightarrow$ H), it is the case that if T is true, H is true. Conceptually, if the truth no longer holds, the meaning has changed. This section reviews work on RTE and how we can use RTE to support our aim to detect significant revision.

RTE task can be either two- or three way output (Sammons, Vydiswaran, and Roth, 2011). The two-way RTE task can have either entailed or not entailed as output, while a three-way RTE task has three types of output: entailed, contradicted, or unknown. Sammons, Vydiswaran, and Roth (2011) provide definitions for each of the textual entailment outputs including examples, which are applicable in the context of text revision. Their examples are as follow, where T is text and H is hypothesis text:

Entail We say that T entails H if the meaning of H can be inferred from the meaning of T.

T The purchase of Houston-based LexCorp by BMI for $2Bn prompted widespread sell-offs by traders as they sought to minimize exposure.

H BMI acquired an American company.

Contradict H contradicts T if a human reader would say that the relations/events described by H are highly unlikely to be true given the relations/events described by T.

T The purchase of Houston-based LexCorp by BMI for $2Bn prompted widespread sell-offs by traders as they sought to minimize exposure.

H BMI bought employee-owned LexCorp for $3.8Bn.

Unknown Reading T and H, the entailment is unknown, which cannot be contradiction.

T The purchase of Houston-based LexCorp by BMI for $2Bn prompted widespread sell-offs by traders as they sought to minimize exposure.

H BMI is an employee-owned concern.

Paraphrase has been shown to be bi-directional textual entailment of the two texts being compared (Androutsopoulos and Malakasiotis, 2010; Μαλακασιώτης, 2011; Romano et al., 2006; Watanabe et al., 2013). The advantage of using RTE in text revision is the directional relationship. Borrowing from the concept of bi-directional entailment for paraphrase detection, for surface change where the meaning does not change, the original and revised texts should have bi-directional entailment. Furthermore, if the truth between the revised sentences no longer holds, we consider the revised sentences as having meaning changed. From computational aspects of identification of significant changes in versioned text documents, we can use this to first filter out those revisions without any meaning change.

There are many approaches used in RTE such as similarity measurement (Tatar et al., 2009), logic (Akhmatova and Molla, 2006), formal semantics (Toledo et al., 2013),

probabilistic approach (Dagan and Glickman, 2004; Sha et al., 2015), including combinations of approaches (Bos, 2014). RTE systems mostly consist of different components (Sammons, Vydiswaran, and Roth, 2011; Magnini et al., 2014; Dagan et al., 2013) such as alignment, inference engine and classification. Our aim of using RTE systems is to evaluate if the revised sentences whether the sentences entail.

There are many components in an RTE system (Sammons, Vydiswaran, and Roth, 2011; Magnini et al., 2014). However, there is no existing RTE system to process revised texts. Hence, instead of comparing different components in RTE systems, we focus on the entailment algorithms. If we regard T and H as revised texts, regardless of the entailment algorithm, the entailment outcome for all of the algorithms are comparable. Examples of entailment algorithms are tree edit distance, transformation- and classification-based (Magnini et al., 2014) which will be reviewed in the subsections below.

2.6.2.1 Tree Edit Distance

The tree edit distance (TED) (Kouylekov and Magnini, 2005) entailment decision algorithm starts by first transforming text, T and hypothesis text, H to the respective dependency trees. A dependency tree or dependency based parse tree is a labeled tree with a one-to-one connection of the word and part-of-speech tags but without phrasal information, an example for a noun phrase, the tag is a noun tag instead of a noun phrase tag. This provides additional linguistic information other than edit operations.

If the RTE evaluation is in the direction of T to H, TED maps the whole content of T to H, using sequence of edit operations, such as insertion, deletion and substitution with each operation having a cost related to it. TED depends on the existing training set to associate the cost and edit operations. There is no existing training set for meaning change categorisation yet.

In this directional approach, T entails H if a sequence of transformation can convert dependency tree of T to the dependency tree of H under certain cost. Inserting a node is attached to the dependency relation of the source label, while deleting a node does not necessarily requires deletion of all its children, rather the children are attached to the parent of the deleted node or a substitution occurs. Only if target node to be substituted has the same part-of-speech as the source node, the node is directly substituted. The relation attached to the substituted node is changed with the relation of the new node. The cost of inserting a word is based on inverse document frequency. Hence for a more frequent word such as a stop word, the cost of insertion becomes 0 while more weight is placed of less frequent words. Deletion cost is 0. Substitution cost relies on the used of dependency based thesaurus.

2.6.2.2 Transformation Based

The aim of the transformation-based entailment algorithm is to apply a sequence of transformations on T to make T identical to H. If the transformation is preserved fully

or partially, T and H preserve the original meaning, hence H can be inferred from T. Magnini et al. (2014) provided an example of transformation: the text is `The boy was located by the police` and the Hypothesis is `The child was found by the police`. In this example, two transformations occur: `boy` $\longrightarrow$ `child` and `located` $\longrightarrow$ `found`. Bar Ilan University Textual Entailment Engine (BIUTEE) is a transformation based EDA (Stern and Dagan, 2014). BIUTEE (Stern et al., 2012) incorporates knowledge-based transformations (entailment rules) with a set of predefined tree-edits other than insert, delete and substitute, in addition to more efficient way of setting the threshold to determine if two texts entails.

2.6.2.3 Classification

This classification based entailment decision algorithm learn a classification model using a maximum entropy (MaxEnt) classifier to combine the outcomes of several scoring functions (Wang and Neumann, 2007). A number of features are extracted at various linguistic levels such as bag-of-words, syntactic dependencies, semantic dependencies and named entities. The scoring functions will calculate the similarity scores of the features. Likewise to the previous EDA, MaxEnt classification EDA depends on an existing RTE training set (i.e. from the third PASCAL RTE challenge - RTE-3 English dataset) too. We explore three different sets of features for revision processing:

1. The most basic set of features (MaxEnt) is bag-of-words (BoW) and lemmas,
2. The second set (MaxEntWNVO) considers the basic features with additional syntactic and semantic dependencies such as hypernym, synonym, part holonym from WordNet (WN) (Miller, 2009) and verb relation of stronger than, can result in and similar from Verbocean (VO) (Chklovski and Pantel, 2004), and
3. The third set (MaxEntAll) considers the second set with additional features: part-of-speech (PoS) and dependency relation or tree skeleton.

We reviewed these three general RTE approaches because there is no existing work that uses RTE for revised sentences or significant revision identification. In the next chapters, we will explore these general approaches for our experimentation.

2.7 Chapter Summary

A taxonomy for revision analysis (Faigley and Witte, 1981) differentiates the changes according to whether the revision alters the meaning of the text or not. According to Faigley and Witte (1981), manual revision analysis includes four categories of revision: formal, meaning preserving, micro- and macro-structure. The four categories are not adequately defined to allow computational implementation.

Goyal et al. (2017) investigates edit importance based on the reviewers, however, edit importance can be subjective between authors and reviewers. This work provided

a general understanding of what affects edit importance. Nevertheless, these works are only limited to scoring edits without being able to automatically classify to the edits according to the taxonomy for analysing revisions (Faigley and Witte, 1981).

There have been computational efforts to categorise various types of text revision (Zhang and Litman, 2015; Daxenberger and Gurevych, 2013; Bronner and Monz, 2012), however none assess the significance of the revision. Zhang and Litman (2015) defined a general pipeline for supervised classification of text revisions in terms of three main processes: revision extraction to automate the extraction of changes in revised texts, revision categorisation for human annotation of different types of revisions, and revision classification as the automated process to differentiate the types of revisions. A few computational components have been identified to help us build the framework to detect significant revision changes: sentence alignment to align revised sentences, paraphrase detection and recognition of textual entailment to evaluate the textual entailment between revised sentences.

The aim of this study is to further examine the impact of revisions made by authors in a multi-author environment. This study aims to propose an approach to classify revisions according to the taxonomy for analysing revision (Faigley and Witte, 1981). In order to build a computational model to detect significant revision in revised text documents, assuming that macro-structure change is a significant change. In order to delve more into the different kinds of revisions, in the next chapter we will present an introspective analysis of a specific versioned text documents: software requirement specification - use case specification. As we aim to create a conceptual model to detect significant changes in revised documents, we have chosen one specialised writing: use case specification and another type: academic drafts.

Chapter 3

A Conceptual Framework for Revision Types Categorisation

The relevant content of the following publications has been integrated into this chapter:

Tan, P. P. , Verspoor, K. & Miller, T. (2015). Structural alignment as the basis to improve significant change detection in versioned sentences. In Proceedings of the Australasian Language Technology Association Workshop 2015 (pp. 101-109).

Tan, P. P. , Verspoor, K. & Miller, T. (2016). Rev at SEMEVAL-2016 Task 2: Aligning chunks by lexical, part of speech and semantic equivalence. In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016) (pp. 777-782).

In Section 2.1.1, the review of Faigley and Witte's (1981) taxonomy of revision analysis considered the suitability of its adoption for revision type categorisation and the possible challenges arising when converting the taxonomy to a computational model to detect significant revision changes. In this chapter, a conceptual framework for revision change categorisation is proposed, with the aim to build a computational framework to identify significant revision. The proposed conceptual framework is derived through investigation and human feedback on versioned text documents in a multi-author environment. This chapter provides the description for the different kinds of revision changes including the definition of significant revision. To the best of my knowledge, there is no existing definition of significant revision change or a system to differentiate revisions based on the impact of meaning change, where major meaning change is considered as significant revision.

If a text, T entails a hypothesis text, H, then a reader reading H, most likely the truth of T can be implied. However for the same T and H, it is not necessary that H entails T. Hence, whether texts entail is directional. The core of our proposed conceptual framework is to assess both the textual entailment concept to determine the type of revision change, which will be described in this chapter.

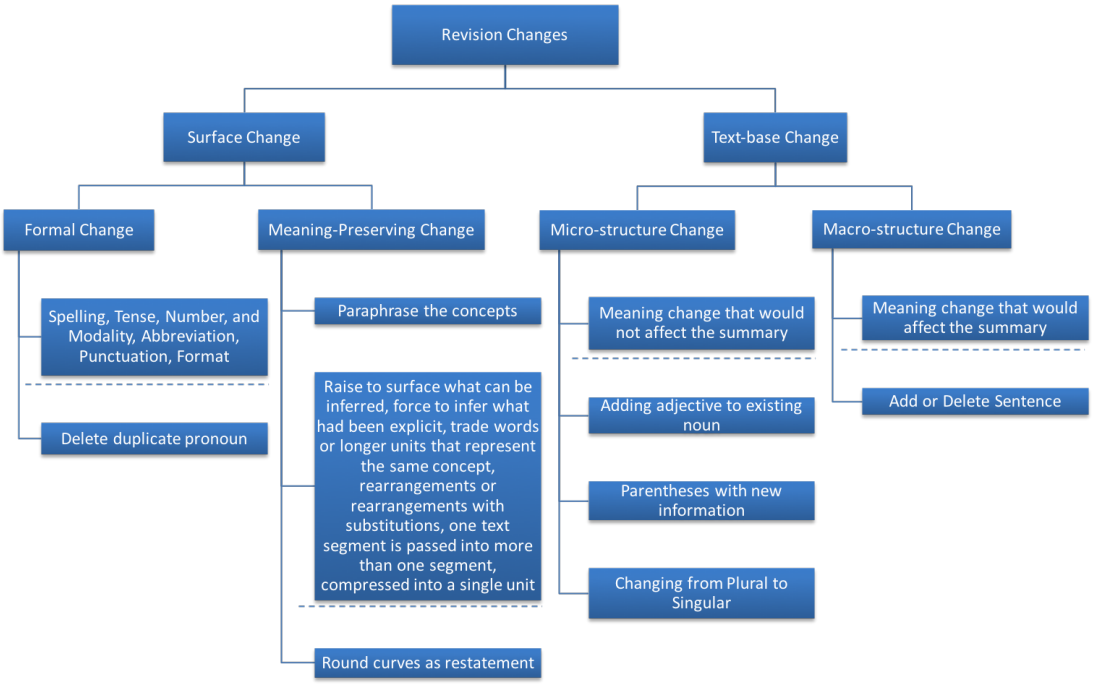
Note: Throughout this thesis, for representation purpose (see Example 3.1.1), the original sentence is denoted as s_o and the symbol $\longrightarrow$ denotes revised to, while the revised sentence is denoted as s_r. For representation of entailment, the symbol used is $\models$.

3.1 An Overview of Revision Types Categorisation Conceptual Framework

This section presents an overview of our proposed conceptual framework to categorise different kinds of revision (Figure 3.1), which has been adapted from (Faigley and Witte, 1981). The main research objective of this thesis is to automate the identification of significant revision changes, other than distinguishing between the different types of meaning change in revision. The revision type categorisation conceptual framework adopts the taxonomy for analysing revision by Faigley and Witte, 1981: formal (FC), meaning preserving (MPC), micro-structure (MiSC) and macro-structure (MaSC), with additional definitions to formalise the taxonomy. Significant revision is considered as MaSC while revision with minor meaning change is MiSC.

In order to distinguish between the taxonomy (Faigley and Witte, 1981) and the proposed definitions in this thesis, the elements above the dotted line in the proposed framework are similar to theirs while the elements below the dotted line are proposed by us (Figure 3.1).

One distinct difference between the taxonomy (Faigley and Witte, 1981) and the proposed conceptual framework in this thesis is the use of bi-directional textual entailment assessment of the revised text to assess the impact of meaning change. Dagan et al. (2013) defined textual entailment as a directional relationship between two texts



Surface Change	Text-based Change	
	Micro-structure Change	Macro-structure Change
$s_o \vDash s_r \wedge s_r \vDash s_o : \{True, True\}$	$s_o \vDash s_r \oplus s_r \vDash s_o : \{True, None\}$	$s_o \vDash s_r \wedge s_r \vDash so : \{None, None\}$

- *where s_o is Original sentence and s_r is Revised sentence*
- *Sentence Meaning can be assessed using Local or Global scope*

Least Significant → Significant

FIGURE 3.1: Revision Type Categorisation Conceptual Framework adapted from Faigley and Witte (1981) by applying bi-directional textual entailment concepts

(text, T and hypothesis text, H), in which the truth of H depends on the truth of T. The concept of bi-directional entailment to evaluate two texts with the same meaning has been applied to paraphrase detection (Tatar et al., 2009). However, rather than applying it in paraphrase detection, when this is adopted in revision change, we propose that the different entailment outcome between the original text, s_o and the revised text, s_r yields a different type of revision change. The concept of *bi-directional textual entailment testing* serves as the core of this conceptual model. The detailed description on the derivation of this approach is presented in Section 3.2.

Another significant purpose of automation in this thesis compared to the taxonomy (Faigley and Witte, 1981) is the formalisation of their taxonomy in a multi-author setting. The automatic identification of significant revision aims to assist the authors

to make better informed decision especially during the transition from one author to another so that authors can focus on revision with meaning change.

Example 3.1.1:

$s_o \longrightarrow s_r$, where

s_o = `I paid a hundred dollars for the tickets to take my family to a movie.`

and

s_r = `I paid a hundred dollars for the tickets, with popcorns and drinks, to bring my family to a movie.`

In our proposed conceptual framework, the impact of meaning change may vary depending on the assessment scope of the revised sentence, either within or beyond that sentence. For instance in Example 3.1.2, s_o has been revised to s_r, where s_r consists of two sentences. In this example, meaning change can occur in the first sentence in the revised form or consideration of meaning change can include additional text beyond the first sentence. Therefore, in the proposed conceptual framework, different *assessment scope* yields a different outcome of meaning change.

Example 3.1.2:

$s_o \longrightarrow s_r$, where

s_o = `I paid a hundred dollars for the tickets to take my family to a movie.`

and

s_r = `I took my family to a movie. I paid a hundred dollars for the tickets.`

Similar to an assessment either within or beyond the revised sentence, assessment scope can be applied to revision at word and phrase levels too. Furthermore, multiple edits can exist within a sentence, hence, for our proposed conceptual model, the focus is on sentence. At sentence level, the two assessment scopes are defined as:

Local The assessment of the impact of change is confined within the revised sentences or the text surrounding the edits but still within the versioned sentence pair.

Global The assessment of the impact of change goes beyond the revised sentence.

The proposed framework must be able to produce the significance between the revision changes. Therefore, another difference between our proposed conceptual framework and the taxonomy is the *scale for impact of change*: from most to the least significant according to this sequence: *Macro-structure Change* > *Micro-structure Change* > *Meaning Preserving Change* > *Formal Change.*

The taxonomy for analysing revision (Faigley and Witte, 1981) divides the revision changes to two main categories: *surface* and *text-base* changes. As our proposed conceptual framework (Figure 3.1) is adopted from this taxonomy where surface change

is revision with no meaning change, and is associated with revision that is least significant compared to text-base change. The sub-categories are similar but additional definitions are introduced to enable computational implementation. The definitions provided below are general definitions, with a few exceptions to these definitions which are listed in Figure 3.1:

Formal Change (FC) Similarly to the taxonomy, this change does not alter the original meaning at all and is generally related to copy-editing changes such as revising the spelling, tense, numbering, and modality, abbreviation, punctuation and formatting. An example of this type of revision is shown in Example 3.1.3.

Example 3.1.3:

s_o = `I paid a hundred dollar for the tickets to take my family to a movie.`

$\longrightarrow$

s_r = `I paid a hundred dollars for the tickets to take my family to a movie.`

Note: For the subsequent definitions, the following original sentence, s_o is used, where $s_o \longrightarrow s_r$:

s_o = `I paid a hundred dollars for the tickets to take my family to a movie.`

Meaning Preserving Change (MPC) Re-phrase or re-word to express the sentence in a different style that does not change the meaning of the sentence and still within the original context. Hence, bi-directional entailment of the revised sentence pair ($s_o \models s_r$ *and* $s_r \models s_o$ *are true*). Revision examples of s_o for local and global assessment scopes are provide in Example 3.1.4 and 3.1.5 respectively.

Example 3.1.4:

s_r = `I paid a hundred dollars to take my family to a movie.`

Example 3.1.5:

s_r = `I took my family to a movie. I paid a hundred dollars for the tickets.`

Formal and meaning preserving changes are grouped together as least significant changes. These two changes are also grouped together in (Zhang and Litman, 2015), although they do not consider the impact of the revision.

Micro-structure Change (MiSC) Revision that alters the meaning of words within the sentence **but** that does **not** alter the overall gist of the sentence in the greater context. This includes addition or deletion of information which does not change the overall meaning. Hence, the revised sentence pair entails at one direction (either original entails revised or revised entails original sentence) and **not** both directions: $s_o \models s_r \oplus s_r \models s_o$. MiSC revisions of s_o for local and global assessment scopes are demonstrated in Example 3.1.6 and 3.1.7 respectively.

Example 3.1.6:

s_r = `I paid a hundred dollars for the tickets, with popcorns and drinks, to bring my family to a movie.`

Example 3.1.7:

s_r = `It was raining heavily last night. I paid a hundred dollars for the tickets to take my family to a movie.`

Macro-structure Change (MaSC) This revision is a significant change as this revision alters the overall gist of the sentence in the greater context. Hence, *no entailment* can be detected between the revised texts. Examples of local and global assessment scopes for MaSC are presented in Example 3.1.8 and 3.1.9 respectively.

Example 3.1.8:

s_r = `We decided to watch movie at home.`

Example 3.1.9:

s_r = `I paid a hundred dollars for the tickets to take my family to a movie. However the movie was canceled due to heavy rain.`

The entailment relations for the different revision types are presented in the table at the bottom of Figure 3.1.

3.2 Inferring Meaning Change in a Text Discourse using Textual Entailment

The taxonomy (Faigley and Witte, 1981) lacks a clear definition for computational implementation, thus, it is important to formalise the different types of revision changes. This section presents how textual entailment is adopted to distinguish the revision types. As stated in the theoretical analysis in Section 2.1.1, differentiation of no meaning and meaning changes can be determined by evaluating whether information added or removed can be recovered or not through drawing inference (Faigley and Witte, 1981). In general language usage, textual entailment is a directional relationship between two fragments of text (Text, T and Hypothesis, H), where T is said to entail H or T $\models$ H if the meaning of H can be inferred from the meaning of T (Pazienza, Pennacchiotti, and Zanzotto, 2005). Textual entailment is never introduced in the taxonomy (Faigley and Witte, 1981). In terms of revision: if the meaning of s_r can be inferred from s_o, then s_o entails s_r or vice versa, then conceptually, the textual entailment approach can be used to detect meaning change in revised texts. Under such circumstance, the texts before and after revision should exist in order to make such assessment. Thus, given an original sentence, s_o and the revised sentence, s_r, meaning change can be determined by evaluating the entailment between s_o and s_r.

From a linguistic perspective, the explanation provided for MPC in (Faigley and Witte, 1981) is clear, in which MPC "includes changes that 'paraphrase' the concepts in the text but do not alter them". In addition, the definitions given by Faigley and

Witte (1981) is based on edit operations such as addition: 'raise to the surface what can be inferred' and deletion: 'force to infer what had been explicit'. These definitions are not directly implementable computationally. Other than the edit operations of distribution and consolidation, the examples provided by Faigley and Witte (1981) are phrase-level instead of sentence-level. However, for classifying revisions, sentence-level is able to capture enough information for higher level revision operations (Zhang and Litman, 2014). Furthermore, sentence-level can accommodate cases where multiple edits are made within a revised sentence which are important to evaluate the overall meaning change. When the definition of MPC is related back to textual entailment, bi-directional entailment has been used in paraphrase detection (Androutsopoulos and Malakasiotis, 2010); two texts are paraphrased means that the two texts entails in a bi-directional manner. With regards to this, in our proposed conceptual framework, not only does MPC has no meaning change, neither do FC. For instance, FC with spelling correction (Example 3.1.3) or meaning preserving revisions that had been rephrased (Example 3.1.4 and 3.1.5), the original meaning of the text remained. Therefore, for surface change which includes formal and meaning preserving changes, $s_o \models s_r$ and $s_r \models s_o$ in our proposed framework are conceptually true. Before fully utilising the concept of bi-directional textual entailment to detect surface changes, further exploration on textual entailment on text revisions is conducted.

The addition of new content or the deletion of existing content is considered as meaning change (Faigley and Witte, 1981). According to Van Dijk (1980), micro-structure in a discourse is the local structure, for instance, sentences and sequence of sentences that include cohesion, anaphora and inference. From this, at the micro-structure level, one sentence should entail the following sentence within a text. Hence, a micro-structure change, can be deduced as a change within the micro-structure level while the summary should remain unchanged as stated in the taxonomy (Faigley and Witte, 1981). Therefore, for a micro-structure revision, either by reading an original sentence s_o, the meaning of s_r can still be inferred but reading s_r, the meaning in the original text can no longer be inferred or by reading s_r the meaning of s_o can be inferred. s_o and s_r cannot be entailed at both ways. An example of micro-structure change where new information is added:

s_o = `I paid a hundred dollars for the tickets to take my family to a movie.` $\longrightarrow$ s_r = `I paid a hundred dollars for the tickets, with popcorns and drinks, to bring my family to a movie.`

In addition to the summary approach as defined by Faigley and Witte (1981), another approach to distinguish between macro- and micro-structure changes is to determine whether the concepts involved in a particular change affect the reading of other parts of the text. In Section 3.1, assessment scope of the sentence is introduced to determine the meaning and the amount of surrounding text to read depends on the sentence itself.

Kintsch and Van Dijk (1978) stated that "A macro-structure must be implied by the (explicit) micro-structure from which it is derived". Hence, within a text discourse,

meaning of sentences at micro-structure entails meaning at macro-structure. Van Dijk (1980) introduces entailment for macro-structure in a text discourse by stating that there is a relationship between the meaning of the text and the topic where each sentence entails (that is, semantically implies) the proposition, in other words, the topics derived from a particular written piece of discourse. Conceptually, if the revised sentences do not entail one another anymore, a macro-structure change has occurred. Assuming now a macro-structure change (MaSC) occurs, $s_o \longrightarrow s_r$, the meaning cannot be inferred from the sentences, in such a way that reading s_o cannot infer the meaning in s_r, neither can reading s_r infer the meaning of s_o. Based on this, examples of macro-structure change:

s_o = `I paid a hundred dollars for the tickets to take my family to a movie.` $\longrightarrow$ `We decided to watch movie at home.`

or

s_o $\longrightarrow$ `I paid a hundred dollars for the tickets to take my family to a movie. However, the movie was canceled.`

In the work by Kintsch and Van Dijk (1978) and Van Dijk (1980), micro- and macro-structures are based on propositions in discourse, where there is no change or revision that occurs, while in the taxonomy (Faigley and Witte, 1981), micro- and macro-structure changes are based on the revised texts. Therefore, we propose to use bi-directional textual entailment testing at sentence-level to categorise text revision according to the meaning change. Thus, in order to evaluate meaning change in a revised texts, it is important the texts before and after revision exist to be able to make inference. Faigley and Witte (1981) stated too, change is to be evaluated sentence-by-sentence, hence textual entailment between revised sentences applies. Our proposed conceptual framework focuses on sentence-level as it needs to be implementable computationally. The summary approach will be considered for future endeavours as computational implementation of this approach requires extensive linguistic understanding of summary in text revision.

In brief, based on the earlier interpretation of micro- and macro-structure in a text discourse, including the concept of textual entailment to infer meaning change in texts, *bi-directional textual entailment testing is proposed for revision type categorisation*, where the *different outcome of the assessment will yield different revision type* as shown in Table 3.1.

TABLE 3.1: Bi-directional Textual Entailment in relation to Revision Changes

Surface Change	**Text-base Change**	
	Micro-structure Change	**Macro-structure Change**
$s_o \models s_r \wedge s_r \models s_o$	$s_o \models s_r \oplus s_r \models s_o$	$\neg(s_o \models s_r) \wedge \neg(s_r \models s_o)$

This basic understanding of bi-directional textual entailment testing and the relationship to the different categories of meaning change in text revision serves as the core to our proposed conceptual framework. The next section, a corpus of specialised

versioned text documents is introduced in order to provide further examine the revisions change category, using examples from this corpus.

3.3 Corpus I: Versioned Use Case Specifications

This section introduces a corpus of revised text documents: use case specifications (UCS), a specific software requirement specification written in natural language. The corpus is versions of UCS for the Orthopedic Workstation (OWS) for Pre-Operative Planning for the Hip. The main purpose this corpus was chosen was because it is an actual revised texts by multi-authors which we can use to investigate what constitute of significant revisions in a multi-author environment.

Multi-author environment, typically involves a myriad of stakeholders in terms of roles and involvement which often leads to multiple revisions of the specification document:

- Setting software requirement by client,
- Design and development of software requirement by system analyst or requirement engineer (i.e. the authors), and
- Review by end user or other stake holders not directly involve in the revision process.

For this specific corpus, there are two versions of the UCS available: version 0.9 and version 1.0, with a total of three authors and introduction of a new author in the later version. Any version that is created right after a version is labelled as *back-to-back* versions. Back-to-back versions have high similarity to each other, but the changes in the later version are significant enough to create another version, which makes these two versions suitable for the task of significant revision change detection. In this work, version 0.9 is labelled as the original version, v_O and version 1.0 is the revised version, v_R. Version 1.0 has been implemented as software in a local hospital.

Similar to most UCS documents, the flows of the software events, pre- and post-conditions, as well as a list of glossary terms used are available. The list of glossary terms contains 27 terms with 11 terms having more than one word. In addition, there are figures and comments of revisions which are disregarded as the focus here is on the direct revision made to the texts.

In this corpus, when comparing the original and revised versions, there are 38 sentences that have no change and 23 sentence pairs with minor edits that could change the meaning substantially. These sentence pairs are called *versioned sentences* and the examples of such sentence pairs are shown in Table 3.3). As observed for this corpus, for versioned sentences, there is a minimum of one edit per sentence pair and a maximum of three edits between the pairs. An edit itself can consist of one or multiple words. Substitution and deletion of words do occur, but a large number of the edits involve adding words in the later version (i.e. 16 out of the total versioned sentence

pairs) with most cases to provide more clarification. These statistics are summarised in Table 3.2.

TABLE 3.2: Changes Statistics for OWS Use Case Specifications: Pre-Operative Planning for Hip Version 0.9 and Version 1.0

Change	**Number of Sentences**	**Number of Words per Sentence**		
		Shortest	Longest	Average
No Change	38	1	50	13
Added/Deleted Sentence	18	4	34	15
Versioned Sentence Pair	23 pairs	2	32	9

The edit operations observed in this corpus correspond to the primitive edit operations identified in (Bronner and Monz, 2012; Faigley and Witte, 1981; Hashemi and Schunn, 2014; Zhang and Litman, 2014). Our research concentrates on the qualitative analysis of back-to-back versions while their work focus on the first and final drafts in which greater differences can be observed between the drafts. It is more challenging to determine the significance of minor edits for versioned sentence pairs in back-to-back versions from semantic perspective as these minor edits often require a particular domain knowledge to comprehend the significance of the change.

The statistics of revision in this corpus (Table 3.2) reveal that there exists standalone sentences, unlike versioned sentence pairs, these sentences have no similar sentence directly associated to them, in other words, full sentences are added or deleted. In comparisons between the original and revised text documents, out of the 18 standalone sentences, only one sentence has been deleted while the rest of the 17 sentences were added in the revised text document. Regardless of the changes, the length or the number of words in the sentences can vary widely (Table 3.2). On average, the number of words per sentence for versioned sentence pairs is slightly smaller compared to standalone revised sentences. When this sentence type is analysed, the shorter sentences in UCS are often in imperative form (as shown in Table 3.3). An imperative sentence is a sentence that gives a command or instruction (Nordquist, 2016). We are required to consider all these in our proposed conceptual framework.

This corpus might not be a large corpus, however it has the criteria of versioned text documents in a multi-author environment, making this corpus compelling enough for qualitative study of the impact of revision changes. The corpus is reflective to show changes for back-to-back versions too. The next section garners feedback from the original authors and non-author participants on categories of meaning change in revised texts.

3.4 Introspective Assessment

This section presents the introspective assessment on the versioned use case specification (UCS) as introduced in the section earlier. An introspective analysis approach provides qualitative assessment of the corpus to further assist in comprehending what

constitute of revision changes and to demonstrate the feasibility of the proposed concepts on an actual versioned text documents. This approach is quite similar to the corpus linguistics method where the corpus is analysed as it naturally occurs (Wallis, 2007) although at smaller scale. The introspective assessment was conducted by the lead author where each of the changes were examined and the impact of meaning change were evaluated.

First, the examination on the revised UCS is performed at the sentence-level, as sentence-level is commonly used in revision works (Bronner and Monz, 2012; Faigley and Witte, 1981; Zanzotto and Pennacchiotti, 2010; Zhang and Litman, 2014). In our proposed framework, *revised sentence*, s is defined as any sentence where word, words or a full sentence has been edited, added or deleted, while *versioned* or *revised sentence pair* is revised sentences that are syntactically similar (s_o, s_r) or for sentence that are added or deleted paired with an empty sentence.

3.4.1 Assessment Scope

In order to demonstrate the different assessment scopes, examples of versioned sentence pairs are extracted from corpus I and presented in Table 3.3. The observation for this corpus is at sentence-level; there are three scopes to assess how the change affects the meaning surrounding the edits:

No change There exist identical sentences with no revision, hence have no impact of change $s_o = s_r$.

Local The assessment of the impact of the change is confined within the revised sentences or the text surrounding the edits but still within the versioned sentence pair.

Global The assessment of the impact of change goes beyond the revised sentences.

3.4.1.1 No Change

For the first sentence pair in Table 3.3, both the versioned sentences are identical with no meaning change. Hence this type of versioned sentence pairs is categorised as *no change*.

3.4.1.2 Local Change

For the second versioned sentence pair example in Table 3.3, the current *diff* approach (as reviewed in Section 2.4.2) extracts out insertion of `OWS, as Annotated X-ray`, deletion of `Information` and insertion of `Record`. Reading the edits alone is not sufficient to understand how much of the meaning has changed. In order to make sense of the edits, readers will read that `X-ray` has been changed to `OWS X-ray` and followed by the as `Annotated X-ray` and the `Patient Information` is substituted with `Patient Record`. `OWS` is the acronym of the system. Although, both `OWS X-ray` and `Annotated X-ray`

TABLE 3.3: Examples of Versioned Sentence Pairs

	Original Sentence, s_o	**Revised Sentence,** s_r
1	Software license checking, if any.	Software license checking, if any.
2	Store X-ray with Current Patient Information.	Store OWS X-ray as Annotated X-ray with Current Patient Record.
3	Calculate Offset of Non-Destroyed Hip.	Calculate Offset of Normal (Contra-lateral) Hip.
4	Select material for Insert.	Select material, internal diameter, and other attributes e.g. low profile, extended rim of Insert.

require auxiliary knowledge to identify and understand the changes, the assertion here is that the assessment of the impact of the changes is confined within these two sentences or the text surrounding the edits but still within the two sentences. When assessment is based on these two sentences alone, the assessment scope is *local*. In discourse representation theory, the local context is the basis of contextual information that is entirely sentence-internal (Kamp, Van Genabith, and Reyle, 2011). Adopting from that definition for revised sentence pairs, local assessment scope is defined as revision changes by which the assessment of meaning change is only confined within the revised sentence pairs.

Nevertheless, the same revision changes presented for local assessment can be assessed beyond the revised sentences. As supported by Kamp, Van Genabith, and Reyle (2011), the argument here is that generally, the contextual information for sentences can be both internal and external.

The second versioned sentence pair example in Table 3.3 is observed again. There is more than one edit in a revised sentence. Multiple edits are common in text revision. If multiple edits occurred within a revised sentence pair, how would one assess the significance of such revision? Taking the second example (i.e. s_o = `Store X-ray with Current Patient Information`, s_r = `Store OWS X-ray as Annotated X-ray with Current Patient Record`), one way of assessing the significance of the revision would be to identify the revision with the highest impact for that pair. Although the revision of `Information` $\longrightarrow$ `Record` is meaning preserving change, addition of `as Annotated X-ray` produced a micro-structure revision because it is an added information which cannot be inferred when reading s_o alone. Therefore, the overall significance of that revision is minor meaning change according to the scale range set for impact of change in our proposed conceptual framework which goes from least to most significant following this sequence: formal change < meaning preserving change < micro-structure change < macro-structure Change (as shown in Figure 3.1).

Rather than evaluating the changes individually, alternatively, the significance of the revision can be assessed using the bi-directional textual entailment testing of the revised sentence pair proposed in the conceptual framework. The textual entailment is evaluated between s_o and s_r at both directions: $s_o \models s_r$ is false however $s_r \models s_o$

is true. According to the proposed revision type category, the revision still results in micro-structure change without needing to evaluate the edits individually. Hence, bi-directional evaluation applicable regardless of the number of edits within the revised sentence.

3.4.1.3 Global

The next change observation, in most cases, are entire revised sentences that are added or deleted that have no matching or similar sentences between the two versions, unlike no change and local change. Mostly, the assessment of impact of change for *global assessment scope* is based on the preceding or/and following sentences, which can be either a revised sentence or an unchanged sentence. Thus, the no change sentences cannot be entirely disregarded because the impact of change for the global assessment might depend on these sentences. Adding and deleting sentence(s) mostly requires global assessment of change. There is a possibility that sentences can be merged together or separated, as a form of sentence re-phrasing, which still retains the same meaning. There is also the case of joined sentences which changes the original meaning of the sentences. Hence, a revised sentence that is assessed using the global assessment scope can be either meaning preserving change, micro- or macro-structure change. The distinction between local or global assessment scope is important to determine the computational approach to assess the impact of change. Local and global assessment scopes are considered in our proposed conceptual framework and the examples to differentiate the two scopes are presented in Table 3.4.

TABLE 3.4: Example of Local and Global Assessments

Sentence	**Example**
Original, s_o	`Label pathology on X-ray.`
Revised, s_r	`Label pathology on Annotated X-ray.` `Predefined Labels includes suggestions.`
Assessment Type	**Example**
Local	`X-ray` $\longrightarrow$ `Annotated X-ray`, still within the revised sentence
Global	`Predefined Labels includes suggestions.`, which refer to the earlier sentence.

3.4.2 Advanced Edit Operation

In the taxonomy (Faigley and Witte, 1981), operations such as permutation, consolidation and distribution exist while in (Zhang and Litman, 2014), these operations are defined as *advance edit operations*, where sentences are merged or separated to more sentences. Using introspective analysis to identify these operations, identification of related sentences can be subjective: a revised sentence can be related to a lot of sentences especially in a multi-author environment where getting agreement among the

authors can add to the challenge of revision text processing. The example in Table 3.4, the sentence `Predefined Labels includes suggestions.` can be regarded as a revision from the previous sentence or is a new sentence added altogether. Under such circumstance, the assumption for cases of permutation, distribution and consolidation is that only the sentence directly precede and/or follow the revised sentence are evaluated. Nevertheless, these cases are not disregarded rather the revision type depends on the assessment scope. Identifying meaning change using local assessment scope alone can be computationally challenging and further incorporating global assessment scopes will be even more challenging as prior knowledge of the relationship between the sentences is required. Local and global assessment scopes are conceptualised in the conceptual framework. However, the computational implementation in this study is limited to local assessment scope.

3.4.3 Bi-directional Textual Entailment

Taking the third sentence pairs from Table 3.3 as example:

s_o = `Calculate Offset of Non-Destroyed Hip.`

s_r = `Calculate Offset of Normal (Contra-lateral) Hip.`

By reading s_o, the meaning of s_r cannot be inferred, however when s_r is read, the meaning of s_o can be inferred:

$s_o \models s_r$: True and $s_r \models s_o$: False, $\therefore$ based our proposed approach, the revision type (s_o, s_r) = micro-structure change.

The observation for this case is as a human reader, `Non-Destroyed` can be referred to `Normal` while an addition of information (i.e. adding `Contra-lateral`) falls under micro-structure revision. Therefore, a computational method that is able to measure the similarity between words and detect that an addition of information has occurred is required. This particular example shows that the advantage when focusing at sentence-level is that there is no not need to consider the assessment scope of the individual edits within the sentences.

Further examination for bi-directional textual entailment using the fourth versioned sentence pair in Table 3.3 as example:

s_o = `Select material for Insert.`

s_r = `Select material, internal diameter, and other attributes e.g. low profile, extended rim of Insert.,`

By reading s_o, the meaning of s_r cannot be inferred, neither can reading of s_r infer the meaning of s_o:

$s_o \models s_r$: False and $s_r \models s_o$: False, $\therefore$ revision type (s_o, s_r) = major meaning change.

Assuming now that `for` $\longrightarrow$ `of` is a grammar correction, the outcome now becomes:

$s_o \models s_r$: True and $s_r \models s_o$: False, $\therefore$ revision type (s_o, s_r) = minor meaning change.

Although `for` $\longrightarrow$ `of` is a formal change, there is added information in s_r, hence the revision type for this pair of revised sentences is minor meaning change, if the changes are evaluated individually. Unlike more obvious grammar errors, this type

of error most likely can only be picked up by the authors. For this case, such revision as grammar correction is feasible but unless explicitly mentioned. On the surface of the revision changes, the intention of the revision cannot be detected. The existence of such case is acknowledged but intention of revision is not within the scope of this research.

Similar to the previous example, the original and revised sentences are extracted and the bi-directional textual entailment are assessed. Therefore, an important thing to note here is computationally, a practical approach to align the versioned sentence pairs and assess the textual entailment of the revised sentence pairs at both directions are required.

For both the third and fourth examples in Table 3.3, although the assessment is presented according to the local assessment scope because these sentences are individual steps in the use case specification, which are inter-related. In other words, the steps can be context specific within the sentence or affect the other steps, thus deleting a step is a major change. Referring back to the work on analysing revisions (Faigley and Witte, 1981), one way of differentiating micro- and macro-structures is to determine if the original summary has changed. As demonstrated earlier, the meaning of sentence can be assessed using local and global scopes which will produce either meaning preserving, micro- and macro-structure changes, especially for cases of anaphora and coercion. This further justifies the idea of whether the concepts involved in a particular change affect the reading of other parts of the text to distinguish between micro- and macro-structure changes might not be an effective approach. This section demonstrates that for most of the revision examples in corpus I, assessing revision at sentence-level does not require the summary approach. Conceptually, the bi-directional assessment of textual entailment still applies at paragraph level, although, the anticipation is that the summary approach might be more precise if the evaluation of meaning change was at paragraph level.

3.5 Human Feedback on Meaning Change in Text Revision

The impact of revision change often varies among authors in a multi-author environment, based on intention and knowledge. Therefore it is essential to correlate the conceptual framework to categorise revision changes to the author's perception of significant revision changes. This section presents the user studies conducted for the corpus described in the previous section (Section 3.3) in order to grasp and formulate a clearer view of significant revision changes. The user studies are separated into:

- Mixture of open- and closed-ended survey questions with the authors (Appendix A).
- Closed-ended questionnaire with non-author participants, which do not know intent (Appendix B).

The purpose for using separate data gathering approaches was to enable authors and non-authors to judge meaning change between revisions while evaluating whether the participants generally agreed with the 4-category meaning change scheme adapted from the taxonomy for analysing revision (Faigley and Witte, 1981). In terms of determining the impact of revision, separating the user studies between authors and non-authors was to evaluate the correlation of judging the impact of meaning change in the revision. The non-author participants did not know the intent of the revisions, hence they could focus on revisions with meaning change.

How authors viewed impact of meaning change remained unknown. The authors were not supplied with an annotation scheme, as to not influence them on the 4-category revision changes and they were given different instructions to observe if there was a variation between meaning change specifically from a language perspective for a specialised versioned text document. Both authors were required to fill in the same survey with one author (labelled as SRSA1) given the freedom to interpret meaning change from an intuitive perspective, although both authors were required to justify their options. The other author (labelled as SRSA2) was requested to evaluate the revision based on meaning change from language aspects. SRSA2 served as a control. The authors' feedbacks were then compared.

For the questionnaires, the participants must not be the author of the versioned text documents. The non-author participants consisted of participants of at least 18 years old and have passed the English language proficiency test for admission to a university degree programme. All participants were presented with the same revision cases as the authors and required to assess the meaning change according to the 4-category meaning change; the participants were asked to rate the revisions, either as formal change, meaning preserving, micro-structure or macro-structure change. As the non-author participants were not directly involved in the revision process, they were supplied with an example for each of the meaning change categories (Table 3.5).

TABLE 3.5: Examples of sentence revision according to revision type as presented in the introductory page of the questionnaire

Original Sentence	I paid a hundred dollar for the tickets to take my family to a movie.
Revision Change	**Example of Revised Form**
Formal	I paid a hundred dollars for the tickets to take my family to a movie.
Original Sentence	I paid a hundred dollars for the tickets to take my family to a movie.
Revision Change	**Example of Revised Form**
Meaning Preserving	I paid a hundred dollars to take my family to a movie.
Micro-structure	I paid a hundred dollars for the tickets, with popcorns and drinks, to bring my family to a movie.
Macro-structure	We decided to watch movie at home.

The revisions extracted were either a full use case or a use case step. Each of the revised cases, before and after revisions were presented side by side in the survey and questionnaire for the participants to evaluate the meaning change. The full survey and questionnaire are available in Appendix A and Appendix B.

The feedback obtained were analysed quantitatively and qualitatively by first comparing the feedback between the authors, then observing the feedback between the non-authors before contrasting their feedback between the authors and non-authors. The analysis is presented in the subsections below.

3.5.1 Authors' Perception of Meaning Change in Text Revisions

Author 1, SRSA1 and author 2, SRSA2 were requested to rate the impact of change for the same revisions they had made with rating:

- 1 for minor change (improvement to style or readability),
- 2 for major change (improvement to style or readability),
- 3 for minor change (meaning change),
- 4 for major change (meaning change), and
- 0 for none

Although all of the cases presented have revisions, the option *none* was provided. From the authors' feedback, none of the authors selected the *none* option. This shows that all changes were considered at either minor or major changes. The authors did not indicate any major change (improvement to style or readability). This can either shows that there is no revision case that falls into major change (improvement to style or readability) or improvement of style or readability is not considered a major change. The authors were prompted on these and their feedback was that they did not consider spelling or grammar correction as major change. Generally, revisions for the purpose of improvement to style or readability should not be major changes. Use case specification for a medical application is highly technical in nature and the initial authors are expert writers, thus, another possibility why none of the authors selected âĂŸmajor improvement in style or readabilityâĂŹ could be these expert writers did not need edits for the purpose of major improvement in style or readability.

We compared the ratings by A1 and A2 for each of the revisions and their ratings are shown in the bubble graph (Figure 3.2). The size of bubble increases with increasing frequency of occurrence. For this corpus, A1 and A2 rated (4, 4) the most frequent, depicting a *larger agreement for major meaning changes*. There are 11 revisions rated as (4, 4) and five out of the 11 revisions have a sentence or sentences added or deleted, for instance, adding `This use case will need to be repeated for each OWS X-Ray loaded for the Current Patient` or deleting `Set IRType for Hip-Replacement`.

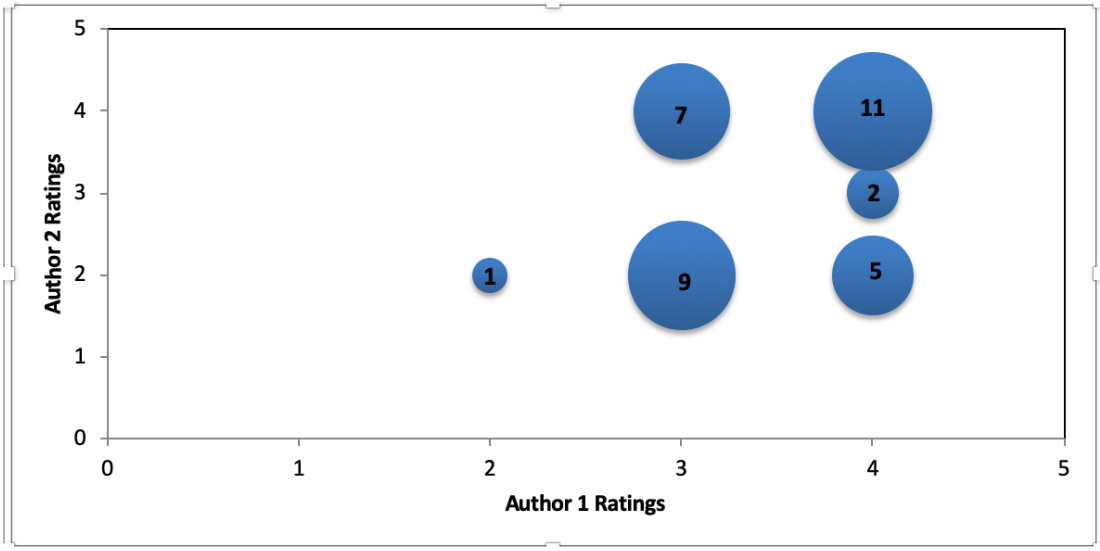


FIGURE 3.2: Author 1 ratings against Author 2 ratings

When revising the same texts, ideally authors should have mostly agreed, if not, with minimal differences in the ratings. However, Figure 3.2 shows low correlation of determining significance of revision in terms of technical and language, with both authors not rating (3,3). For this corpus, it is unlikely that there is no revision with minor meaning change as some of the revisions were rated 3 by either one of the authors (Figure 3.2). Rather, as shown in Figure 3.2, A1 rated one-scale lower or higher than A2 or i.e. (3,2), (3,4) and (4,3) with the highest occurrence at (3,2), demonstrating that for the same revisions, A1 tends to judge the revision as meaning change. A particular note here is the cases of (4,2), in other words, for the same revisions, one author rated as meaning change, while another rated as no meaning change. Overall A1 judged higher significance, mostly 3 and 4. In order to illustrate this possibility, an example of revisions that had been rated as (4, 2) by the two authors is extracted: `Current Patient Information` ⟵ `Current Patient Record`. When the justifications for the significance of rating are referred, A1 stated that "this is describing an artefact, rather than a vague (and potentially incorrect) collection of data" while the justification provided by A2 is "more specific, should align with the glossary (for the UCS)". In brief, both authors agreed that the change is to be specific about the term used, however when judging the impact of change, "describing an artefact" leads to major meaning change while "align to the glossary" leads to improve to style and readability. These can be an indicator that both the authors have different perspectives, where one author has the tendency to weigh the impact of change higher.

We performed detailed analysis of the difference between the ratings by A1 and A2 and presented in Figure 3.3. The ratings by A1 and A2 fall into one of these categories: no difference, one-scale or two-scale differences, where one-scale is defined as either one of them selected meaning preserving change or minor meaning change, or in another case, either one of them selected minor meaning change or major meaning

change. Two-scale difference is defined as either one of them selected meaning preserving change or major meaning change. In order to present the rating differences between A1 and A2, A1 rated two-scale lower than A2 is symbolised using $A1 \ll A2$, A1 rated a scale lower than A2 is symbolised using $A1 < A2$, A1 and A2 rated the same or $A1 = A2$, A1 rated one-scale higher than A2 or $A1 > A2$ and A1 rated two-scale higher than A2 or $A1 \gg A2$.

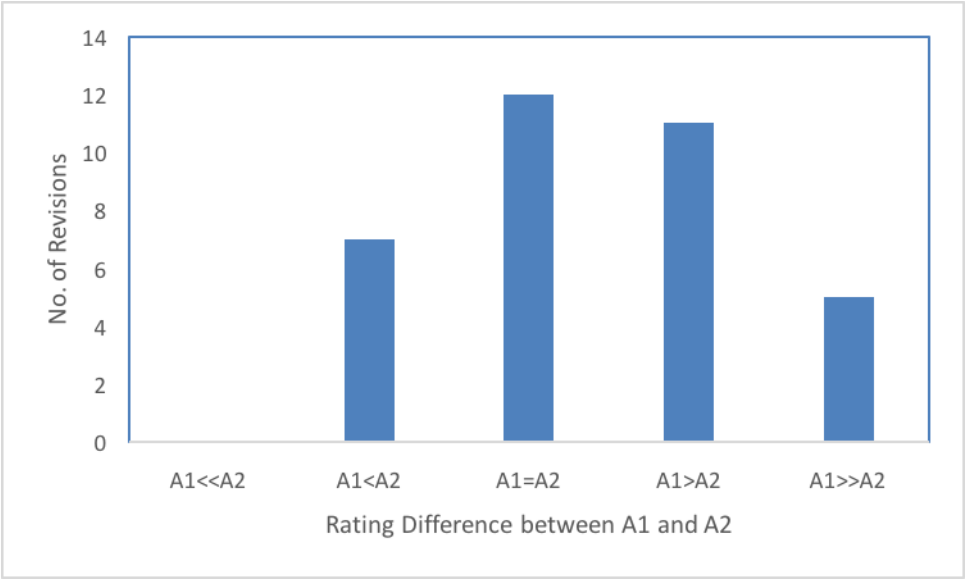


FIGURE 3.3: The difference in significance ratings between A1 and A2: A1 rated two-scale lower than A2 (A1 $\ll$ A2), A1 rated a scale lower than A2 (A1 < A2), A1 and A2 rated the same (A1 = A2), A1 rated one scale higher than A2 (A1 > A2) and A1 rated two scale higher than A2 (A1 $\gg$ A2)

Generally, *A1, rated higher impact of change compared to A2*, where A1 rated 16 out of the 35 revisions with higher impact compared to Author 2, A2, while A2 rated 7 out of 35 revisions with higher impact of change compared to A1 (Figure 3.3). Most of the differences occurred in the category A1 rated one-scale higher than A2, which shows that for specialised writing, in this case, use case specification (UCS), the author had the tendency to rate the impact of change more significantly compared to the same revisions being evaluated for the impact of change based on language aspects. A1 was given the freedom to rate the revisions according to how impact of change was normally rated for SRS or technicality of changes in SRS, whereas A2 was specifically informed to rate based on the language aspects. This could be that when change is perceived technically, the change has more impact in comparison to the same change perceived linguistically, as observed:

- A1 rated mostly 3 and 4, and
- the only 2 by A1 was agreed with A2

In brief,

1. Overall A1 rated revision as more significant compared to A2. Perception affected how author rate the changes.

2. The rating (4,4) indicated that MaSC was agreeable.

3. MiSC was not agreeable with mostly one scale difference, perceived either as MPC or MiSC.

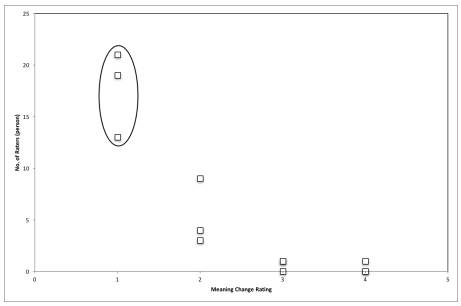
Our proposed conceptual framework will consider at the language level rather than technical level so that the framework is applicable in general. MiSC for our proposed framework will be defined in Section 3.5.2.

3.5.2 Authors versus Non-authors' Perception of Meaning Change in Text Revision

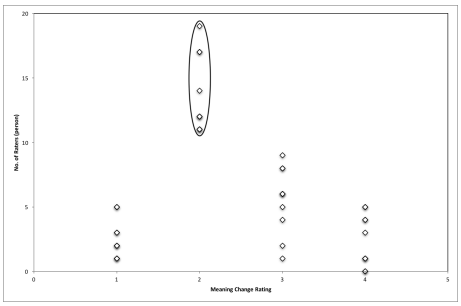
In order to further demonstrate judgement on impact of change based on language aspects, non-authors participants were presented with the same revision cases as the authors and requested to rate the impact of change based on the language aspects. As the participants had never seen the revisions, the participants had to rely on language aspects to evaluate the meaning change. They were required to select one of the four categories of meaning change. As with the authors, they were provided with examples of revision for each of the four meaning change categories. Their ratings were observed and presented in Figure 3.4.

There was a total of 24 non-author participants. When we analysed each of the revisions, some revision cases had a majority of the participants selecting a category. Figure 3.4 (a) - (d), *some revisions have obvious meaning change category* as majority of the participants selected the obvious category. There were three revision cases observed with the number of participants that selected FC the highest (Figure 3.4 (a)), a square shape is used to represent this distribution of ratings and the majority selection is circled). The majority here is as high as 21 out of the 24 participants selecting FC, while for the same revision case, there is no participants that selected MaSC. This shows that some FC revisions were more obvious such as spelling mistakes, which was the case. Further observation on the distribution of ratings with the majority ratings circled, some cases of MPC (distribution represented using diamond shape in Figure 3.4 (b)), MiSC (distribution represented using triangle shape in Figure 3.4 (c)) and MaSC (distribution represented using'x' symbol in Figure 3.4 (d)) were more obvious.

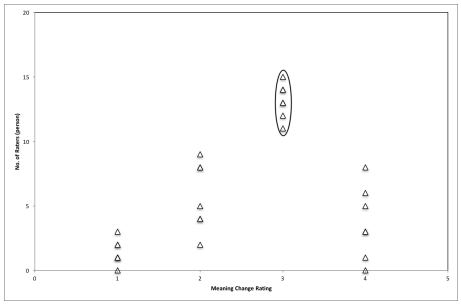
For the cases where a majority of the participants selected MaSC (Figure 3.4 (d)), when compared to the authors' feedback for the same revision cases, there are three revision cases that showed agreement between the authors and majority non-authors. All three of these revision cases were either adding or deleting a sentence. Recall that authors mostly agreed for macro-structure changes (Figure 3.2). Although all the six revision cases selected by the majority as MaSC were either additions or deletions



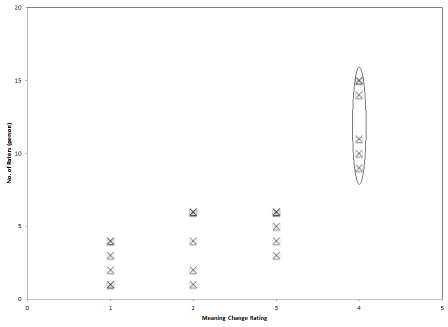
(a) Majority indicated Formal Change



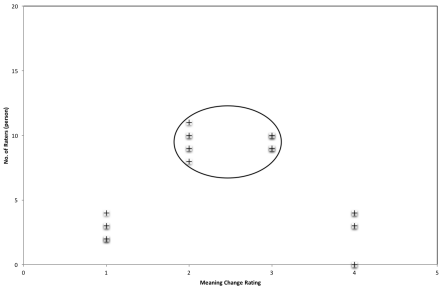
(b) Majority indicated Meaning Preserving Change



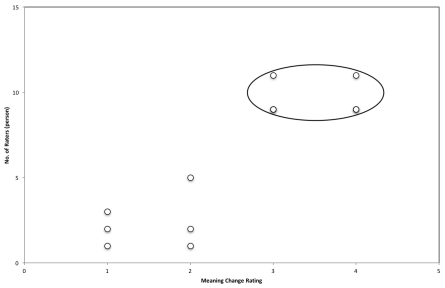
(c) Majority indicated Micro-structure Change



(d) Majority indicated Macro-structure Change



(e) Boundary Cases: Meaning Preserving and Micro-structure Changes



(f) Boundary Cases: Micro-structure and Macro-structure Changes

FIGURE 3.4: Revisions as Rated by 24 Non-Authors categorised according to majority selection for each revision type and boundary cases

of a sentence, however, not all the revision cases presented with added or deleted sentences will result in macro-structure change.

When the overall distributions for two categories are observed, there are revision cases with divided selection or no obvious one majority category selected by a majority of the participants. The difference between the two categories is one scale difference. These revisions are labelled as boundary cases. There are two obvious pairs of categories that fall into boundary cases: meaning preserving and micro-structure changes (Figure 3.4 (e) with the cross symbol representing the distribution of ratings) and micro-structure and macro-structure changes (Figure 3.4 (f) with the circle representing the distribution of ratings). An example of revision for the boundary cases of MPC-MiSC is substitute: `Destroyed` $\longrightarrow$ `Diseased`, in the original sentence: `Identify Replacement Parameters of Destroyed Hips`. Not only did the non-authors rated

this revision as MPC and MiSC, for this revision case, A1 rated 3 while A2 rated 2. When the authors justifications are referred, A1 stated that "Not sure if it is a meaning change or just using better terminology" although A1 rated as 3, while for A2, the justification for the rating was because "change to standard medical terminology only". Hence, for this revision case, both authors agreed on the change for improvement legibility. Within the same corpus, there is another quite similar revision: `destroyed hip` ⟶ `diseased (ipsi-lateral) hip`. Majority of the non-authors rated MiSC for this revision case, rather than the boundary case of MPC-MiSC. However, a majority of non-authors viewed addition of `(ipsi-lateral)` as addition that changed the meaning.

There are three revision cases that fall into the boundary case of MiSC-MaSC, where all three of the revision cases involved adding a sentence. Based on these three revision cases, two of the cases had both authors rated as MaSC, while the other, A1 rated MiSC and A2 rated MaSC; depicting boundary cases too. Recall the analysis for the cases of majority participants that selected MaSC (Figure 3.4 (d)), all of the cases were either adding or deleting. Therefore, for the boundary case of MiSC-MaSC, as A2 generally rated based on language aspects, for this corpus, adding a sentence is likely to be a major meaning change.

Specifically in this corpus, there is no boundary case where FC-MaSC or MPC-MaSC, although there are cases rated as (4,2) or (MaSC, MPC) by the authors. Language wise, the probability that a particular revision had both meaning and no meaning changes should be low. The most probable justification here is that non-authors and A2 evaluated the impact of change based on language aspects, while A1 rated did not. Clear categorisation by non authors means categorisation is understood by non authors as well, just that some cases are unsure whether MPC-MiSC and MiSC-MaSC.

As author 1 (A1) and author 2 (A2) only agreed on a few revision cases (i.e. rated (2, 2) or (4, 4)), the revision cases they agreed on are plotted against the majority ratings for those revision cases and is presented in a bubble chart (Figure 3.5). Those cases where authors could not agree are ignored. The bubbles in the bubble chart in Figure 3.5 are focused on the upper left of the chart, which demonstrate that authors generally have a tendency to rate impact of change as more significant (i.e. (1, 2), (1, 4), (2, 4) and (3, 4)). The bigger the bubble, the more revisions are rated by the authors and non-author participants where the biggest bubble is (3,4). The bubble graph also shows where authors and non-author participants agreed upon is (4, 4). Nevertheless, more data or a different type of corpus is required to make any assertive claim.

3.6 2-Category and 4-Category Meaning Change in Text Revisions

The existing taxonomy for analysing revision (Faigley and Witte, 1981) starts with two-category (2-category) meaning change (i.e. surface and text-base changes), which

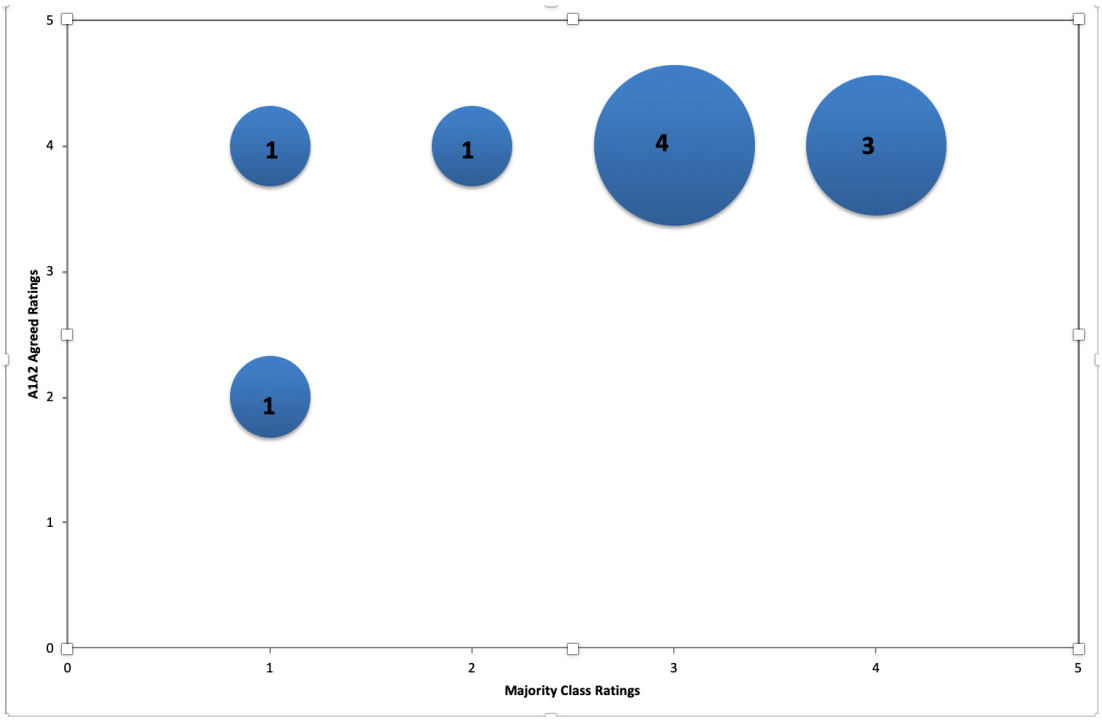


FIGURE 3.5: Majority versus Authors' ratings

branch out to four-category (4-category) meaning change (i.e. formal, meaning preserving, micro- and macro-structure changes). Even though in the user studies 4-category meaning change is presented, for analysis purposes, 4-category and 2-category meaning change classification framework are used, in line with the objective to comprehend how authors and non-authors perceive the different categories of meaning change. For the 2-category meaning change classification framework, the feedback gathered for formal change and meaning preserving change are collapsed into no meaning change (or surface change) while the feedback for micro- and macro-structure changes are collapsed into meaning change (or text-base change). The 2-category meaning change classification approach is similar to works by Bronner and Monz (2012) and Zhang and Litman (2015) where they considered higher level 2-category classification before finer-grained categories.

Based on the 2-category and 4-category meaning classification framework, the inter-rater reliability measures are calculated between authors (Table 3.6) and among the non-author participants (Table 3.7). A brief explanation of inter-rater reliability measures is provided in Literature Review chapter (Section 2.5.1).

Inter-rater reliability is typically used to measure whether the raters agree with each other according to a rating or annotation scheme (Gwet, 2014) (review on inter-rater reliability measurement is in Section 2.5.1). The inter-rater reliability measures we obtained are low (Table 3.6 and 3.7). As for our user studies, an annotation scheme has yet to be fully developed at this stage, thus, the measurements might not be entirely representative. Rather, the qualitative analysis in Section 3.5 is more reflective of the authors' and non-authors' judgment on meaning change in text revision.

TABLE 3.6: Inter-rater reliability measurements from the feedback of two authors'

	4-Category	**2-Category**
N agreement	13	21
N disagreement	22	14
Simple Agreement (%)	37.1	60.0
Scott's π	-0.065	-0.134
Cohen's κ	0.098	0.075
Krippendorff's α		
Nominal	-0.004	-0.118
Ordinal	0.32	-

TABLE 3.7: Inter-rater reliability measurements for feedbacks from the Non-author Participants

	4-Category	**2-Category**
average pairwise percent agreement	38.5%	60.5%
Fleiss' κ	0.153	0.209
average pairwise Cohen's κ	0.161	0.22
Krippendorff's α		
Nominal	0.154	0.21
Ordinal	0.266	-

Based on the authors' feedbacks obtained, the simple agreement for 4-category meaning change is 37.1% while 2-category meaning change is 60% (Table 3.6), similarly for non-authors, the simple agreement for 4-category is 38.5% while 2-category is 60.5% (Table 3.7). *Authors and non-authors have higher agreement on no meaning and meaning change (2-category) compared to lower level categories of meaning change (4-category).* This is also demonstrated through Fleiss' κ and average pairwise Cohen's κ for 2-category is greater compared to 4-category (i.e. Fleiss' κ: 0.209 > 0.153 and Cohen's κ 0.220 > 0.161) (Table 3.7). 2-category is expected to have higher agreement because it is identified as either with or without meaning change. The authors and non-authors were never informed to categorise based on 2-category change. Instead, better agreement on 2-category meaning change can suggest observation be done to first filter based on 2-category meaning change prior to the sub-categories. This is translated as part of our computational implementation which will be explained in the next chapter.

From the feedback, non-authors have higher inter-rater reliability measurements compared to authors (Table 3.6 and 3.7). Thus, the 4-category meaning change is still valid. Krippendorff's alpha, α (Formula 2.10) is an inter-reliability measurement which considers disagreement between raters (Krippendorff, 2011) with four different types of calculation: nominal, interval, ordinal and ratio. Nominal type treats each of the categories as singular category, interval type treats the categories as quantitative values, while ordinal type treats the categories as in an order form and ratio type treats each of the category as a ratio to another. In the case of revision categories

(i.e. formal, meaning preserving, micro- and macro-structure changes), the nature is nominal and not likely to be in the form of interval or ratio. However, based on the feedback obtained, both authors and non-authors observed higher $\alpha_{ordinal}$ compared $\alpha_{nominal}$. Hence, the *revision types can be viewed as ordinal* where formal revision has the least impact of change, gradually increasing to meaning preserving change, follow by micro-structure revision, with macro-structure revision as the highest impact of change. This is incorporated into the conceptual model and an important part of deciding *macro-structure revision as significant change*.

In brief, the user studies provided supporting data specifically for specialised versioned texts in a multi-author environment, demonstrating how authors and non-authors perceived meaning changes. The lesson learned from this round of user studies is extended to develop a guideline for annotating meaning change in text revision, which will be explained in detailed in Chapter 5. The next section investigates into the applicability of this our framework through introspective analysis.

3.7 Preliminary Comparison - Similarity and Alignment

Earlier sections present human analysis of the revised sentences. In this section, using revised sentence pairs in versioned use case specifications, we conducted preliminary computational comparison of similarity measurements (reviewed in Section 2.3.2 and 2.3.3). We also compared alignment of different types tokens between the revised sentences by representing change with word error rate (WER) (reviewed in section 2.3.1.2). The purpose of these comparison is to observe the correlation (reviewed in Section 2.3.4) to the impact of change (i.e. none for surface change, minor for micro-structure change and significant for macro-structure change) as rated by the authors and non-authors (see Section 3.5 for detailed analysis of the human feedback). *Similarity measurements have inverse correlation to human feedback on significance*; the higher the similarity values, the least significant the changes are, similarly to semantic similarity. *Semantic similarity has stronger inverse correlation to human feedback on significance compared to string similarity*. WER is shown to correlate better with human feedback. The last column of Table 3.8 shows the correlation between the output of the approaches and human feedback on significance of the revision. Although different corpus and rating systems are used, Table 3.8 compares the aspects we considered to Goyal et al. (2017).

When Table 3.8 is referred, semantic similarity have inverse correlation to impact of change and alignment using word and glossary terms are helpful in alignment revised sentence pairs. Even though this comparison shows that considering more factors correlate better to edit importance, Goyal et al. (2017) did not consider minor and major meaning changes. Furthermore, based on our analysis on actual revisions by the authors, authors and non-authors (i.e. reviewers) rated the impact of change differently (see Section 3.5). We focus on task of categorising revisions considering the different factors for computational processing. This preliminary comparison also shows that

TABLE 3.8: Comparison of various approaches to support identification of significant changes with the correlation coefficient against human feedback on significance

. ED - Edit Distance, LM - Lexical Meaning, SI - Syntactic Information, NE - Named Entities, RS - Flesch Kinkaid readability scores, CC - Correlation Coefficient, WER - Word Error Rate

Approach	ED	LM	SI	NE	RS	Measure-ment	CC
Similarity, SIM							
String	✓					SIM	r = -0.34
Semantic		✓	✓			SIM	r = -0.59
Alignment							
Word	✓					WER	r = 0.63
Phrase	✓		✓			WER	r = 0.58
Word + Glossary Terms		✓	✓			WER	r = 0.66
Edit Importance Scores, EIS (Goyal et al., 2017)							
Edit Importance	✓		✓	✓	✓	EIS	ρ = 0.979

a standard evaluation measurement and labelled corpus are required for the task of significant revision identification so that direct comparison can be made for different approaches proposed for the task (Chapter 5).

3.8 Derivation of the Different Kinds of Revision Changes

Basically, the 4-category and 2-category meaning changes are based on the taxonomy for analysing revision (Faigley and Witte, 1981), however, these categories are not directly computationally implementable. This section presents the outcome of the qualitative analysis on individual revision changes with regards to the responses by authors and non-author participants, from a broader usage of the language such as general pattern of revision that falls into a certain category. From this analysis, additional definitions for the different types of revision with the related entailment outcome are included and summarised in Table 3.9.

As presented in the literature review (Chapter 2), the current change detection feature in text editors and paraphrase approaches cannot fully support categorisation of different types of revision changes. The significant revision identification does not only identify macro-structure change, the proposed conceptual framework includes identification of formal, meaning preserving and micro-structure changes. There can be errors such as wrongly identified cases of formal change that are labelled as meaning preserving change or an error in significant revision change detection where a macro-structure change is wrongly identified as meaning preserving change. Having

TABLE 3.9: Different kinds of revision changes based on feedback by human with the related entailment outcome

Meaning Change Category	Revision Observation	Entailment Outcome
Formal	• delete redundant pronoun • subject verb agreement correction	s_o entails s_r, s_r entails s_o
Meaning Preserving	• restatement within round brackets or parentheses • similar word or phrase substitution	s_o entails s_r, s_r entails s_o
Micro-structure	• add extra information to existing sentence such as adding a Noun Phrase, description, adjective • confirming what is not	s_o entails s_r but s_r does not entail s_o or s_o does not entail s_o but s_r entails s_o
Macro-structure	• add new information (add new sentence(s))	No entailment between s_o and s_r

many surface changes identified as meaning change will risk the significant revision changes as unhelpful in a multi-author environment. Defining the different kinds of revisions is crucial in derivation of the significant revision changes framework not only to prevent the incorrect categorisation outcome but concise definitions for each of the categories that can lead to an automated approach in identifying significant revision (Figure 3.1).

3.8.1 Formal and Meaning Preserving Changes

When further examination of each of the revisions presented was made, one observation was for revision case `which must be` $\longrightarrow$ `must be`, where majority of the non-author participants considered this as formal change (54.2%). Linguistically, this change is analysed as a *deletion of redundant pronoun*, which falls under formal change even though (Faigley and Witte, 1981) stated that the "reader is forced to infer what had been explicit" which falls under meaning preserving change. For this specific revision, when the authors' ratings are referred, both regard this as improvement to style or readability or no meaning change. Thus, for this type of revision, the revision type is formal change. This form of revision, the higher level categorisation will be no meaning change and if wrongly categorised between formal or meaning preserving

change, the risk is lower compared to an actual significant revision but categorised under no meaning change category.

Another case of revision which majority of general participants selected as formal change is *spelling correction to eliminate 's'*. The same revision case: `X-rays` ⟶ `X-ray`, repeatedly occurred but at different use cases all through the specification. The same revisions occurred twice at different use cases are extracted for the participants to evaluate too. The purpose of presenting the same revisions is to check for consistency among the participants. For both the cases, a majority of participants (more than 79%) agreed as formal change, which is consistent with Faigley and Witte's (1981) formal change - spelling correction. On the contrary, the revisions were repeated all through the specification by the authors, as not intended for spelling correction, instead to revise from *plural form (i.e. x-rays) to singular form (i.e. x-ray)*. This provided an explanation why the authors had selected the revisions as meaning change. As mentioned earlier, the intention of revision is beyond the scope of in this thesis although this example clearly demonstrate that intention can produce different outcome of the meaning change.

When these revisions are observed at the sentence level, the original and revised sentences have the same meaning. Hence, when we evaluated the entailment outcome, the original sentence, s_o and the revised sentence, s_r, both sentences entailed each other.

3.8.2 Micro-structure Change

According to Faigley and Witte (1981), punctuation is defined under formal change. In corpus I, one of the revision cases with punctuation observed involves additional pairwise round brackets or parentheses with enclosed word(s) within. Observe carefully, round brackets with words within are in actual fact two revisions; adding the brackets and adding the word(s). If the *word(s) in the curve brackets refers to the same thing but just as a re-statement for clarification*, for instances, `medial points of` ⟶ `medial points (or tops)` and `outer edge of` ⟶ `outer edge (cortex) of`, the majority of the participants selected meaning preserving. The authors agreed with these two revisions as improvement to style or readability. Faigley and Witte (1981) stated that addition in meaning preserving change is "raise to the surface what can be inferred", which in this case, re-statement for clarification. On the contrary, if the revision is within the *parentheses to add new explanation*, for example - ⟶ `(one pair just inferior to the lesser ...)`, then this type of revision falls under micro-structure change as selected by majority of the general participants. Language wise, these classifications are consistent with the intended use of parentheses in the English language where the parentheses either "encloses information for clarification, or set aside from the main point" (Straus, Kaufman, and Stern, 2014). Hence, revisions with curve brackets can yield either meaning preserving or micro-structure change as indicated by the majority participants. As a human reader, differentiating the usage of brackets

might seem like a trivial task however computationally, a lot of aspects need to be factored into in order to reach the same deduction as a human.

Faigley and Witte (1981) stated abbreviation was listed as formal change. The revision cases observed in this corpus are acronyms, which is a form of abbreviation. One example from this dataset is that the acronym is placed in round brackets `IRType` $\longrightarrow$ `IRType(AS)`. This type of revision falls back to the earlier parenthesis example and not just adding acronym; this revision case is judged based on the original word(s) of the acronym contained within the round brackets. The general feedback provided by the participants rated this revision as a meaning preserving change while the authors on the other hand evaluated it as meaning change because of the specialised meaning of the acronym. In another revision case of adding an acronym to existing an noun: `X-ray` $\longrightarrow$ `OWS X-ray`, the full words are referred to when making judgement of the revision type. Although this specific revision is repeated in the user studies for consistency check, there is no majority standing for this kind of revision because 'OWS X-ray' is a specific terminology which causes confusion: does adding OWS raise what can be inferred already or is it adding an adjective which changes the noun? For this type of revision, the authors' ratings are used: micro-structure change. When contrasted with `X-ray` $\longrightarrow$ `annotated X-ray`, adding the term 'annotated' changes the description of the original noun although the noun remained the same. This kind of revision is considered as micro-structure change because nothing can be inferred whether the x-ray is annotated or not.

Other than raising what can already be inferred, the majority confirms with the definition in the taxonomy (Faigley and Witte, 1981): "trade words or longer units that represent the same concept" as meaning preserving change. Majority participants agreed that *substitution of word or phrase* leads to meaning preserving change `information` $\longrightarrow$ `record`, `non-destroyed` $\longrightarrow$ `normal`. In such case, bi-directional textual entailment applies, though detection of synonymous words or phrase is helpful too.

In the taxonomy, meaning change is defined as "whether new information is brought to the text or whether old information is removed in such a way that it cannot be recovered through drawing inference" - where no new information is regarded as *surface change* while *text-base* change involves adding of new content or the deletion of existing content. The distinction between micro-structure and macro-structure changes is not clear. When the feedback are analysed further, one observation for revisions that fall into micro-structure change category is the revision have the same right noun `surgeon authentication` $\longrightarrow$ `authentication`. Another example is deletion of `at least`, which a majority regards as micro-structure change because old information cannot be derived anymore. This is consistent with the introspective assessment of the corpus (Section 3.4), that for micro-structure change, the original and revised sentences entail one way but not both ways, although entailment was not introduced before in the taxonomy (Faigley and Witte, 1981).

3.8.3 Macro-structure Change

A majority of the participants regard that adding a sentence or more falls under macro-structure revision. Van Dijk (1980) described an example of macro-structure in a discourse is topic. However, we propose that at sentence-level for macro-structure revision, revised sentence pairs have no entailment: original sentence does not entail revised sentence, likewise, revised sentence does not entail original sentence. Thus, if a sentence is added or deleted without a revised sentence pair, this sentence is paired with an empty sentence, producing a macro-structure change in this proposed conceptual framework.

Even though there is only one revision case: `destroyed hip` $\longrightarrow$ `diseased (ipsi-lateral) hip`, a majority of participants rated this as micro-structure change. Notable, `destroyed hip` $\longrightarrow$ `diseased hip` falls under meaning preserving change while adding `(ipsi-lateral)` is added information which cannot be inferred from the previous state. This revision case is evaluated as two revisions where micro-structure change (i.e. adding information) is more significant than meaning preserving change (i.e. replacement of similar word). Alternatively, based on our proposed bi-directional textual entailment testing approach, this too is a micro-structure change.

This analysis shows that identifying the revision change type is very much dependent on the linguistics aspects of the revision. This will post a challenge computationally because it requires both syntactic and semantic understanding of the revision: current computational approaches either measure semantics at individual word level or syntactic differences between sentences.

3.9 Chapter Summary

This chapter presented our proposed conceptual framework (Figure 3.1) adapted from taxonomy for analysing revision (Faigley and Witte, 1981) to categorise revision according to meaning change to four revision types: formal, meaning preserving, micro- and macro-structure changes. In order to propose this conceptual framework, we performed introspective assessment on an actual versioned text document, conducted user studies with the authors and non-authors and analysed their feedback. We propose to work at sentence-level. Based on our understanding from existing literature, the core in our proposed conceptual framework is bi-directional textual entailment testing of revised sentences where different outcome of the textual entailment will yield different revision type (Table 3.1). Revised sentence pairs that entail will most likely be no meaning change, while revised sentence pairs that totally do not entail will most likely be macro-structure revision. Micro-structure change has original sentence and revised sentence entail at one direction only.

Before this, how authors perceived revision changes especially the significance remained unknown and what constitute of revision changes was not well defined. Through introspective analysis of the corpus we introduce in this chapter, the context

for revised sentences can be assessed locally (sentence-internal) and globally (sentence-external) with different scopes of assessment producing different impacts of revision change, for the same revision change.

Inter-rater reliability measure of Krippendorff's alpha α we obtained indicated that the authors and non-authors viewed the 4-category revision types as ordinal, establishing the scale of impact of change from formal change <meaning preserving change <micro-structure change <macro-structure change. Macro-structure change is significant revision. In order to categorise revision change, one possible way is higher level meaning change (2-category) must first be evaluated (i.e. meaning and no meaning change) before considering the sub-categories.

The scope of this conceptual model is limited to sentence only to develop understanding on the detection for text revision based on the representation. The main aim of this thesis is to computationally be able to detect significant revision changes between revised text documents. Conceptual framework itself does not equate directly to computationally implementable. Hence, using this proposed conceptual framework, steps have been taken to transform this to a computational model, which is presented in the next chapter. In ensuring that the computational model generally applies, the proposed computational model will be evaluated using another corpus of versioned text documents. Both the computational model and a different corpus will be elaborated in detail in the next chapters respectively.

Chapter 4

Significant Revision Identification Computational Framework

A computational framework using a natural language approach has been proposed in Chapter 3. This conceptual framework utilises textual entailment evaluation to differentiate the revisions according to meaning change: formal, meaning preserving, micro- and macro-structure changes, with micro-structure change considered as minor meaning change while macro-structure change is considered as significant revision change. *This chapter focuses on developing the conceptual framework to a computational framework*: the transition from the taxonomy for analysing revision (Faigley and Witte 1981) to the conceptual framework and finally to the computation framework is summarised in Figure 4.1. In a nutshell, various approaches (reviewed in Chapter 2) that can support computational implementation of our proposed conceptual framework are explored and the computational framework are implemented in phases.

4.1 Overview of Significant Revision Identification Computational Framework

The section presents an overview of our proposed computational framework. Our proposed computational framework derives from our proposed conceptual framework, which identifies meaning change based on the taxonomy for analysing revision (Faigley and Witte, 1981) by adopting textual entailment evaluation to support Van Dijk (1977)'s concept of micro- and macro-structure in written discourse. Table 4.1 summarises the core concepts of our proposed conceptual framework, which serve as the foundation to design a computational framework for significant revision identification. Our proposed computational framework (Figure 4.2) consists of three phases: versioned texts pre-processing (Section 4.2), textual entailment evaluation (Section 4.3), and revision type categorisation (Section 4.4).

Overall, this framework inputs two versions of a text document, (v_o, v_r), where v_o is the original text, while v_r is the revised version of that text, and outputs the revision type, E_m for each of the revised sentence pairs, $(s_o, s_r)_m$ extracted from (v_o, v_r), where s_o is original sentence, s_r is revised sentence of s_o and m is the total number of revised

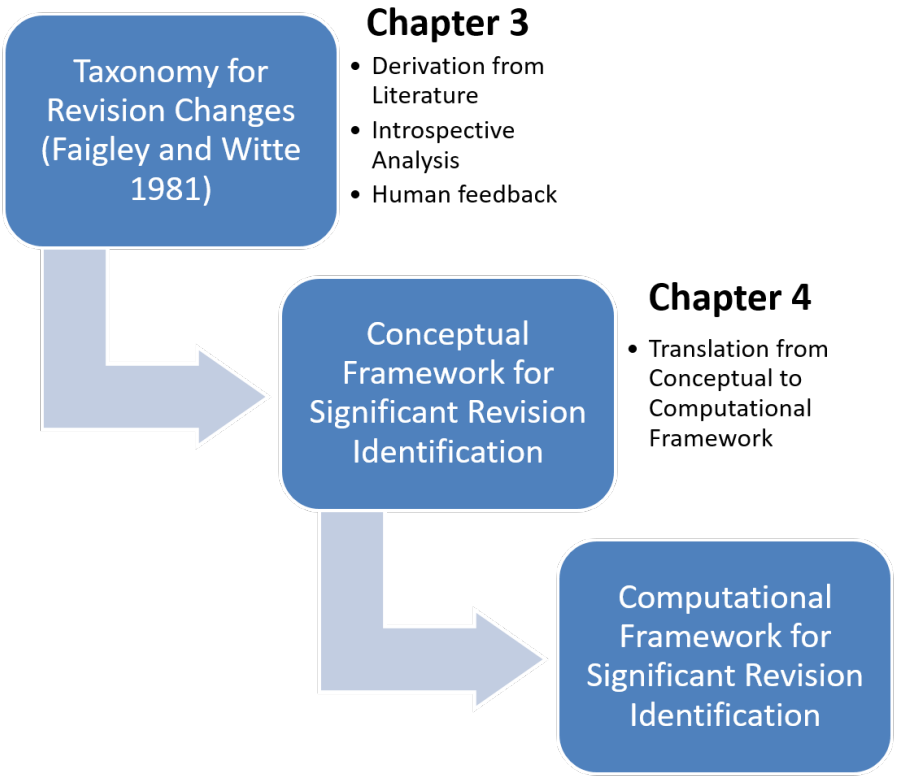


FIGURE 4.1: The process of developing Significant Revision Identification from Taxonomy to Computational Framework

sentences. On the whole, this computational framework is applicable to any two versions of a text document, thus, this framework is independent of the versioning or collaborative writing tools. An author can make multiple revisions and commit these revisions. Regardless of how many times the same author revises the text, the version committed right before being passed on to the next author is considered as v_o. Similarly, the next author can make multiple revisions. The latest version revised by this author is considered as v_r. In the case where authors commit right after one another, v_o and v_r are considered as *back-to-back versions*. Hypothetically, in a multi-author environment, information regarding the changes made is useful for the transition of ideas throughout the revision process.

TABLE 4.1: Core in the conceptual framework for significant revision identification

Surface Change	**Text-base Change**	
	Micro-structure Change	**Macro-structure Change**
$\{s_o \models s_r \wedge s_r \models s_o\}$ = {true, true}	$\{s_o \models s_r \oplus s_r \models s_o\}$ = {true, non}	$\{s_o \models s_r \wedge s_r \models s_o\}$ = {non, non}

Each of the phases has different purpose, input and output. A brief description for each of the phases is as followed:

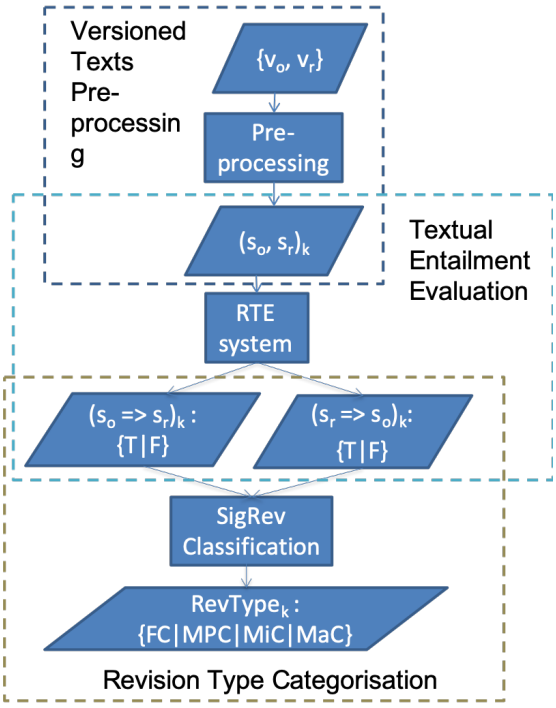


FIGURE 4.2: Significant Revision Identification Computational Framework

versioned text documents pre-processing This phase inputs two versions of a text document (v_o, v_r). This is a pre-processing step that compares (v_o, v_r) and extract versioned sentence pairs based on the *diff* output. This phrase outputs a list of m revised sentence pairs, $\{s_o, s_r\}_m$.

textual entailment evaluation The revised sentence pairs, $\{s_o, s_r\}_m$ from the previous phase serve as input to this phase. This phase evaluates the entailment of the revised sentences within the versioned sentence pairs, $(s_o, s_r)_m$ at both the relational directions: $(s_o \models s_r, s_r \models s_o)_m$ where the entailment outcome can be either true (i.e. entail) or false (i.e. non entail). This phase outputs the entailment outcome for each of the revised sentence pairs, $(s_o \models s_r, s_r \models s_o)_m$.

revision type categorisation The outcome of textual entailment evaluations from the previous phase serves as input to this phase, $(s_o \models s_r, s_r \models s_o)_m$. This phase categorises the revised sentence pairs to one of the four types of revision changes (i.e. formal, meaning preserving, micro- or macro-structure) according to the bi-directional textual entailment testing rules (Table 4.1). However, the rules only categorise the sentences to surface change (i.e. no meaning change), micro-structure change and macro-structure change. An additional component in this phase differentiates the sentence pairs that have been categorised as surface changes to formal and meaning preserving changes. The output for this phase, which is also the overall output of our proposed computational framework is the revision type for each of the revised sentence pairs, E_m.

An example for a pair of revised sentences that goes through the processes in our proposed computational framework, $s_o \models s_r$ as true and $s_r \models s_o$ as true. Consequently, the entailment outcomes for this sentence pair are assessed according to the rules in Table 4.1 to determine the revision type for that particular versioned sentence pair. For this pair of example , where $\{s_o \models s_r, s_r \models s_o\} = \{T, T\}$, hence the revision type for (s_o, s_r) is meaning preserving change. This process is repeated for all of the versioned sentence pairs extracted for the versioned texts (v_o, v_r).

4.2 Versioned Texts Pre-processing

The first phase of our proposed computational framework for significant revision identification is versioned text documents pre-processing (Figure 4.3). The inputs to this phase are two versions of a text document (v_o, v_r) and outputs a set of versioned sentence pairs extracted from (v_o, v_r), $(s_o, s_r)_k$, where k is the number of revised sentence pairs extracted from (v_o, v_r). We will first explain the mechanism for versioned text pre-processing.

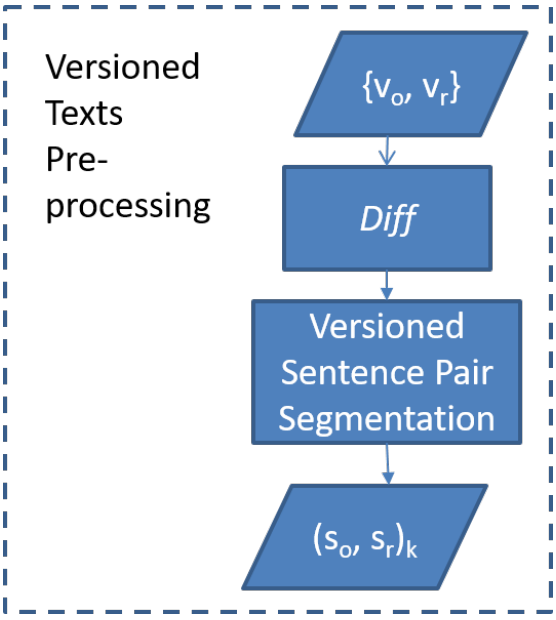


FIGURE 4.3: The process flow in Versioned Texts Pre-processing Phase

Versioned text documents (v_o, v_r) are first pre-processed using *diff* utility (review on *Diff* utility in Section 2.4.2). *Diff* utility compares the differences between v_o and v_r, producing what has been added or deleted between the two texts. The texts used for our experiments are LaTeXfiles, hence in our implementation, LaTeX*diff* [1] is used. The original and revised LaTeXtexts (v_o, v_r) are inputted into the LaTeX*diff*. It produces a LaTeXfile with additional tags of add - for texts that had been added into v_r and delete - for texts which had been deleted from v_o (examples with add and delete tags are shown in Example 4.2.1 and 4.2.2).

[1] https://3142.nl/latex-diff/

The target of next process is to extract the versioned sentence pairs from the output of the *diff*. We propose an algorithm (Algorithm 1) to segment and extract the versioned sentence pairs based on the output of *diff* utility. In order to produce versioned sentence pairs from the *diff* output, first, the *diff* output is segmented at sentence-level, SS_k which includes the add and/or delete tags as shown in Example 4.2.1 and 4.2.2 which are extracted from Figure 4.4. Segmented sentences that do not involve revision or are part of any add or delete scope are ignored, so are citations, tables and figures, as the focus is on revised sentences.

Example 4.2.1:
First segmented sentence, $SS_1 =$

```
‘‘\DIFaddbegin \DIFadd{The underlying motivation for this work is to
identify user characteristics which may be useful for improving
information access over threaded discourse, using the particular example
of forum data.}\DIFaddend’’
```

Example 4.2.2:
Second segmented sentence, $SS_2 =$

```
‘‘\DIFaddbegin \DIFadd{There is}\DIFaddend
\DIFdelbegin \DIFdel{This work is part of ILIAD , an ongoing effort to
improve information access in linux
forums.}\DIFdelend’’
```

Based on add and delete operations, there are generally six types of segmented sentences considered in our algorithm. The example for each type is provided in Table 4.2.

Segmented sentence from *diff* output,

$$SS_k = (m - word)_n + \delta_p^q + \alpha_i^j \tag{4.1}$$

where k is number of extracted out segmented sentence, $k > 0$,
$(m - word)_n$ is contiguous unchanged m-word; $m \geq 1, n \geq 0$, if no change, $n = 0$,
δ_p^q is contiguous deleted q-word; $q \geq 1$, $p \geq 0$, if there is no deletion, $p = 0$, and
α_i^j is contiguous added j-word; $j \geq 1$, $i \geq 0$, if there is no addition, $i = 0$.

The sequence of added, deleted and unchanged contagious words is dependent on the segmented sentence. For each of these segmented sentences, s_o and s_r are extracted according to the sequence of the added, deleted and unchanged contagious words. If the tag is add, α^j is added to s_r, while if the tag is delete, δ^q is added to s_o. If SS_k contains contiguous unchanged words, the words are added to both s_o and s_r. The process continues to check if it is the end of SS_k while continuing to form s_o and s_r according to the rules as above. Each SS_k produces a set of (s_o, s_r). For cases where

TABLE 4.2: Segmented Sentence from *diff* output

Segmented Sentence Type	Example
Full Add	`\DIFaddbegin \DIFadd{The underlying motivation for this work is to identify user characteristics which may be useful for improving information access over threaded discourse, using the particular example of forum data.}\DIFaddend`
Full Delete	`\DIFdelbegin \DIFdel{Our contribution to the project is techniques to identify characteristics of forum users, building on earlier work in the space.} \DIFdelend`
Partial Add and Partial Delete	`\DIFaddbegin \DIFadd{There is}\DIFaddend \DIFdelbegin \DIFdel{This work is part of ILIAD, an ongoing effort to improve information access in linux forums.}\DIFdelend`
Partial Add only	`Inter-annotator agreement, \DIFaddbegin \DIFadd{ based on KendallâĂŹs $\tau$ and associated p-value }\DIFaddend`
Partial Delete	`We present a definition of four basic user characteristics and an annotated dataset, \DIFdelbegin \DIFdel{ which will be made publicly available.}\DIFdelend`
Partial Add and Partial Delete	`\DIFaddbegin \DIFadd{To address this,}\DIFaddend we have designed a set of attributes that we expect to be helpful in improving information access over \DIFdelbegin \DIFdel{threaded discourse.}\DIFdelend`

s_o or s_r is a full added or deleted sentences, the sentence is paired with null sentence.

The algorithm to extract s_o and s_r from SS_k is shown below (Algorithm 1).

Data: Segmented sentences, SS_k
Result: Pairs of versioned sentences, $(s_o, s_r)_k$
for *each E* **do**
  s_o = empty sentence ;
  s_r = empty sentence ;
  while *not at end of E* **do**
    read current tag;
    if *add* **then**
      read j-word ;
      s_r += j-word ;
    end
    if *delete* **then**
      read q-word ;
      s_o += q-word ;
    end
    if *unchanged* **then**
      read m-word ;
      s_o += m-word ;
      s_r += m-word ;
    end
  end
end

Algorithm 1: Algorithm to extract Versioned Sentence Pair, $(s_o, s_r)_k$ from segmented sentence, SS_k of *diff* Output between two versioned of a text document, v_o and v_r

We observed two possible cases that might cause our proposed algorithm to not effectively. When the output LaTeXfile is run using a LaTeXprogram such as TeXstudio [2]: an output as shown in Figure 4.4 is produced. The first case is during revision, where authors might just add or delete certain points and may or may not realised it, the sentences can be incomplete (as depicted in Figure 4.4). During the extraction process of s_o and s_r, the segmentation might not necessarily produce full sentences. For instance, in this sample, SS_2 will produce s_o = `There is` and s_r = `This work is part of ILIAD, an ongoing effort to improve information access in linux forums.` The second is due to the output of *diff* might not necessarily produce nicely "diffed" sentences such as wrong start and end of a sentence pair because *diff* treats punctuation just like any other character. Although our algorithm is able to extract the sentence pairs, the sentences are incomplete. This is an advantage using textual entailment evaluation as the text might not necessarily involve complete sentences and authors do not need to adjust their way of revising to use our proposed approach.

The output from this process is represented in a list of k versioned sentence pairs: (k, s_o, s_r). Based on Figure 4.2 as input, an example of sentence pair extracted using

[2] https://texstudio.org

~~The underlying motivation for this work is to identify user characteristics which may be useful for improving information access over threaded discourse, using the particular example of forum data. There is~~ This work is part of ILIAD (Baldwin et al. 2010) , an ongoing effort to improve information access in linux forums. Our contribution to the project is techniques to identify characteristics of forum users, building on earlier work in the space (Lui 2009) . The problem that we face here is two-fold: Firstly, there is no established ontology for characteristics of forum users~~, so~~ . To address this, we have designed a set of attributes that we expect to be helpful in improving information access over ~~threaded discourse. However~~forum data. Secondly, in order to ~~utilize these characteristics in information access, we need a volume of annotated data that would be infeasible to obtain by manual annotation~~exploit user characteristics we would need to evaluate a large number of users. This quantity of data would be much too large to be processed manually. We therefore apply supervised machine learning techniques to allow us to effectively ~~annotate large quantities of forum data~~ discover the characteristics of a large number of forum users in an automated fashion.

FIGURE 4.4: Sample Output of LaTeX*Diff* between the original text, v_o and revised text, v_r. Red strike off shows deletion, while blue curly underline shows addition, black text is unchanged text

our proposed algorithm is presented in Table 4.3.

TABLE 4.3: Example of sentence pair from segmentation process

Example	SS_6 = `addStart{To address this,} we have designed a set of attributes that we expect to be helpful in improving information access over delStart{threaded discourse.}delEnd`
Output	6, `we have designed a set of attributes that we expect to be helpful in improving information access over threaded discourse., To address this, we have designed a set of attributes that we expect to be helpful in improving information access over`

However, in order to be input for textual entailment system, we require the format of text, T and Hypothesis, H. According to our proposed textual entailment evaluation, each of the sentence pairs will be evaluated directionally, i.e. $s_o \Rightarrow s_r$ and $s_r \Rightarrow s_o$. Hence, for each sentence pair, we produce pairs of T and H: (T = s_o, H = s_r) and (T = s_r, H = s_o). The overall output of versioned text preprocessing phase is pairs of (T, H), where $(T, H)_k = \{(s_o, s_r), (s_r, s_o)\}_k$.

4.3 Textual Entailment Evaluation Phase

Our proposed core concept requires the textual entailment outcome between the revised sentence pairs in order to determine if the change is a FC, MPC, MiSC or MaSC. Hence in the textual entailment evaluation phase (Figure 4.5), we utilise the recognition of textual entailment (RTE) system to recognise whether T entails H, for (T, H) = $\{(s_o, s_r), (s_r, s_o\}$, which are the input to this phase.

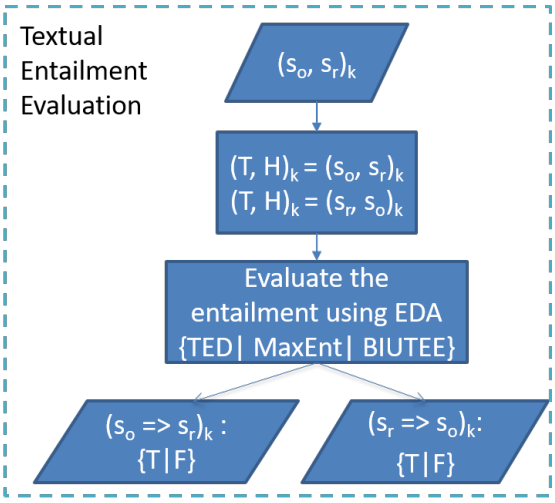


FIGURE 4.5: The process flow in Textual Entailment Evaluation Phase

Currently there is no textual entailment approach or RTE system designed specifically to evaluate revised sentence pairs as proposed by us. Thus, an existing open source RTE system is chosen: the Excitement Open Platform (EOP) for Textual Inferences System (Magnini et al., 2014) to allow evaluation of the proposed approach. The workings of RTE in EOP system are stated in (Magnini et al., 2014). Hypothetically, any RTE system that is able to recognise textual entailment of two texts can be used in our proposed framework.

RTE systems are complex with multiple components such as preprocessing, enrichment, alignment, classification and main decision making approach (Sammons, Vydiswaran, and Roth, 2011; Dagan et al., 2013). We focus on the main decision making approach or specifically the entailment decision algorithm (EDA) in a RTE system that can best support bi-directional textual entailment evaluation of revised sentence pairs. Entailment decision algorithm is the process to compute an entailment decision for a given Text, T and Hypothesis, H pair (Magnini et al., 2014). This phase explores different entailment decision algorithms in the RTE system for the purpose of recognising whether the texts entails one another in revised sentence pairs. For experimentation, three main EDAs are explored: tree edit distance based (Kouylekov and Magnini, 2005), transformation based (Stern and Dagan, 2014) and classification based (Wang and Neumann, 2007). Classification EDA is shown to be effective for RTE in English language (Magnini et al., 2014).

Tree edit distance (TED) based EDA is similar to the current change detection feature which focuses on edit operations: addition, deletion and modification but with the addition that the sentences are converted to dependency trees, hence using the part-of-speech (PoS) tags. For transformation based EDA, this EDA applies a sequence of transformations such as synonymous words to transform T to H while preserving the meaning to check if the meaning of H can be inferred from T. The third comparison approach is classification based EDA. Based on the training set, various scoring functions or linguistics features such as syntactic dependencies, semantic dependencies and name entities that are extracted for T and H. Detailed explanation on the EDAs are provided in Section 2.6.2).

Briefly, the inputs to this phase are pairs of T and H from the previous phase, which are revised sentence pairs. These pairs of T and H are inputted into the EOP system evaluating $s_o \models s_r$ and $s_r \models s_o$ as shown in Table 4.4.

TABLE 4.4: Inputs to RTE System

Directed Relation	T	H
$s_o \models s_r$	s_o	s_r
$s_r \models s_o$	s_r	s_o

The overall output of this phase is the textual entailment decision from the RTE system for each of the revised sentence pairs at different directions and is represented as (k, true/non entail, true/non entail), where k represents the pair of (s_o, s_r), the second parameter is the textual entailment result in the direction of s_o to s_r, while the third parameter is the textual entailment result in the reverse direction (i.e. s_r, s_o). EOP system is able to produce three sets of output: (true, true), (true, non entail) and (non entailment, non entail). An example of input and output for this phase is provided in Table 4.5. The bi-directional textual entailment results will be used as input to the revision type categorisation phase.

TABLE 4.5: Example of Input and Output for RTE Phase

Input	6, `we have designed a set of attributes that we expect to be helpful in improving information access over threaded discourse., To address this, we have designed a set of attributes that we expect to be helpful in improving information access over`
Output	6, entail, entail

4.4 Revision Type Categorisation Phase

The revision type categorisation phase is where the output from the bi-directional textual entailment is utilised to determine the revision types of FC, MPC, MiSC and MaSC. The input into this phase: $(s_o \models s_r, s_r \models s_o)$ = {(true, true), (true, non entail), (non entailment, non entail)}. According to our proposed conceptual framework (Table 3.1), bi-directional entailment evaluation, surface change or no meaning change (entails at both directions), micro-structure meaning change (entails only at one direction) and macro-structure meaning change or significant change (non entailment). Hence, an additional processing is required to distinguish surface change as either FC or MPC. Thus, there are two components within this phase: bi-directional textual entailment evaluation and differentiation of formal and meaning preserving changes (Figure 4.6). The mechanism for both the components are explained in the subsections below.

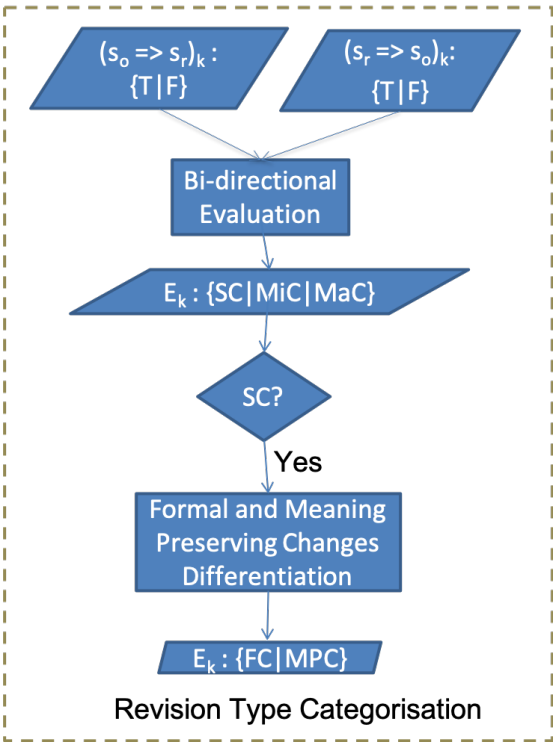


FIGURE 4.6: The process flow in the Revision Type Classification Phase

4.4.1 Bi-directional Textual Entailment Evaluation Component

The input to this component is the textual entailment judgment between the sentences within the revised sentence pairs at both directions: $(s_o \models s_r, s_r \models s_o)$ = {(true, true), (true, non entail), (non entailment, non entail)}. This particular component uses a rule based categorisation approach. This component is compares $s_o \models s_r$ and $s_r \models s_o$ to produce the revision type, E according to the rules in Table 4.1. The comparisons are, for each of the revised sentence pairs, k,

if $(s_o \models s_r, s_r \models s_o)$ = (true, true), then E_k = SC,
if $(s_o \models s_r, s_r \models s_o)$ = (true, non entail), then E_k = MiSC.
if $(s_o \models s_r, s_r \models s_o)$ = (entail, non entail), then E_k = MiSC.
if $(s_o \models s_r, s_r \models s_o)$ = (non entail, non entail), then E_k = MaSC.

The output of this phase is represented as a list of (k, surface/micro-structure/macro-structure). An example of input and output to this component are provided in Table 4.6.

TABLE 4.6: Example of Input and Output for Bi-direction Entailment Evaluation Component

Input	6, entail, entail
Output	6, surface

4.4.2 Surface Change: Differentiation between Formal and Meaning Preserving Changes

This component starts by filtering out all of the revised sentence pairs that entailed both ways or the output of 'surface' from the previous component, (k, surface) where k is the k-revised sentence pair labelled by the bi-directional entailment phase as surface change. The remaining revised sentence pairs take the value assigned from the previous components which are micro-structure and macro-structure changes or E_k = {micro-structure, macro-structure}.

For each of the surface change sentence pairs, the k-versioned sentence pair, $(s_o, s_r)_k$ is retrieved. The string similarity between s_o and s_r is calculated using Jaro-Winkler (Winkler, 1990) is used to calculate the string similarity measurement (reviewed in Section 2.3.2.1). Formal change such as spelling corrections, formatting and grammar errors while paraphrases are considered as meaning preserving change. Thus, formal change will have higher lexical overlap compared to meaning preserving change. Therefore string similarity ≥ 0.9 is set for formal change and any value below that is considered as meaning preserving change.

The input and output for this component are presented in Table 4.7.

TABLE 4.7: Example of Input and Output for Formal and Meaning Preserving Change Differentiation Component

Input	6, surface
Output	6, meaning preserving

4.5 Chapter Summary

This chapter discussed the derivation of computational framework (Figure 4.2) based on our proposed conceptual framework (Figure 3.1) to enable the automatic significant revision identification between versioned text documents. The translation focuses on translating the core of our proposed conceptual framework which is to test the bi-directional entailment outcome of the revised sentence pairs to categorise them according to the revision types. The computational framework has three main phases namely versioned texts pre-processing to identify versioned sentence pairs (s_o, s_r), textual entailment evaluation to recognise the textual entailment between the revised sentence pairs and the revision type categorisation to distinguish the revision types based on the bi-directional textual entailment outcome of the revised sentence pairs. Overall, the computational framework inputs two versions of a text documents and output the revision type for each of the revised sentence pairs as formal change, meaning preserving change, micro-structure change or macro-structure change.

The next step is to evaluate the proposed computational framework. The framework has to be tested on an independent corpus other than the one used to propose the conceptual framework (Section 3.3). Hence, in the next chapter, another type of versioned text document will be introduced.

Chapter 5

Development of Comparison Data and Baseline Comparison

In the previous chapter, the task of significant revision identification that categorise revised sentence pairs to one of the revision types (i.e. formal, meaning preserving, micro-structure and macro-structure changes) is introduced. However, there is no existing annotated corpus or even an annotation guideline that can be used to evaluate our proposed task. This chapter focuses on development of evaluation data and baseline comparison.

Due to limited versioned text documents which fulfil the criteria of a corpus, this study specifies the guidelines for the construction of a corpus where an actual versioned text documents revised by multiple authors is developed. The development of this corpus includes human annotation of the revision types. This corpus will serve as the basis for further development of the corpus to a full dataset in the future. In this chapter too, an analysis of the inter-annotator reliability measure of the human annotation is provided.

As there is no existing approach for comparison, the current track changes feature is considered as the baseline comparison for the task of significant revision identification. The current track changes feature in text processor presents changes that have been added or deleted, which is similar to Levenstein edit distance (LvD) calculation (reviewed in Section 2.3.1.1). The correlation between the annotated data by human and edit distance is observed, then followed by the development of our proposed baseline approaches based on character- and word-level LvD.

5.1 Corpus II: Drafts of Academic Papers

The task of significant revision identification (SigRevId) is defined as follow: given two versions of a text document (v_o, v_r), with each version revised by different authors, the significance of the revised sentences is identified according to one of the four categories: formal, meaning preserving, micro-structure and macro-structure changes. However, there is no existing annotated resource of academic papers that is available for analysis of significant revision identification. Therefore, there is a need to identify such drafts in order to construct a suitable corpus. In particular, available paper drafts

in a multi-author environment, with at least one author must be able to annotate for all the versions of the paper. This corpus is referred to as Corpus II, and will be used for evaluating the computational framework.

Academic papers typically undergo series of revisions, each revision also known as draft, by multiple authors prior to the final version being published. The selection of drafts of academic papers with two or more authors serves as an appropriate case study for revisions in multi-author environment. Three papers with multiple revisions by various authors are selected for this corpus with each of the papers labelled as P1, P2 and P3 respectively. The identification of revision is focused on texts that had been added, deleted or modified, or versioned sentences only, not comments and changes to figures and tables.

Although an author can revise a text document multiple times, only the last draft before passing to the next author are considered in this corpus. The computational framework compares two versioned text documents to calculate the total number of sentences before and after each round of revision (Section 4.2). Every versioned text documents is given a version ID, where smaller ID number denotes earlier version and vice versa. For instance, P1, among the two authors, a total of 20 drafts are committed, but only seven drafts are selected for comparisons in this corpus. For each round of revision, for example in P1, version IDs 1136 and 1144, where 1133 is an earlier version committed by one author, 1144 is the later version committed by another author. The total number of sentences before and after revisions and the number of revised sentences for each of the revisions are summarised in Table 5.1.

TABLE 5.1: Corpus Summary for drafts of Academic Papers

Paper ID	Versions			
P1	$d_{1136,1144}$	$d_{1150,1155}$	$d_{1167,1168}$	$d_{1168,1170}$
N Sentences (before → after)	163 → 156	162 → 178	158 → 157	157 → 151
N Revised Sentences	38	32	3	22
P2	$d_{1782,1801}$	$d_{1801,1806}$	$d_{1806,1808}$	$d_{1808,1809}$
N Sentences (before → after)	143 → 150	150 → 128	128 → 131	131 → 133
N Revised Sentences	8	91	19	10
P3	$d_{3428,3436}$			
N Sentences (before → after)	157 → 151			
N Revised Sentences	132			

Table 5.1 shows P1 and P2 were revised back-to-back four times, while P3 was revised once. The total number of sentences, before and after revisions, depends on the revision by the authors, thus is independent of the total or rounds of revisions. For example, the first revision for P1 (i.e. $d_{1136,1144}$), the total number of sentences is reduced from 163 to 156 and involved 38 sentences being revised that encompasses addition, deletion and modification of sentences.

5.2 Human Annotation of Significant Revisions

Having reliable human annotation of revision types is important because the annotated data is used to compare to the revision types output by the computational approach. Through this comparison, we will be able to evaluate which of our proposed computational approaches performed better and analysis can be performed to examine the weaknesses and strength of the methods. There is neither existing annotated data for the task of SigRevId, nor existing annotation guideline for categorising revised sentences to one of the revision changes. As learnt from human annotation (Section 3.5), there is a need for a refined annotation guidelines for SigRevId due to the discrepancy in human annotation which would limit the reliability of SigRevId:

- The participating authors tended to weigh the revisions more significant compared to non-authors.
- Annotators did not like the previous presentation of the revisions.
- The texts other than the revised texts do matter in identifying the significance of the revisions. Hypothetically, for significant revision identification, revised sentences are better representation.

This section presents the refinement of the previous human annotation on significant revisions (Section 3.5); creating annotation guidelines to improve the annotation process for human annotation on drafts of academic papers.

5.2.1 Annotation Guidelines

Annotation guidelines provide the human annotators with a general understanding of the task. The definition and examples for each of the revision types are provided in the annotation guidelines (Appendix C). Current track changes feature which present characters and words being added, deleted and modified. Rather than presenting the revisions side by side as presented in Section 3.5, for annotation of drafts of academic papers, similar presentation as current track changes feature, LaTeXdiff is used to show characters and words that had been added, deleted and modified between two drafts. We believe the revised sentence pairs are more helpful for the readers to identify the significance of the revision hence, other than presenting the edits, the original and revised sentence pairs are highlighted.

The annotation guidelines (Appendix C) consist of four main sections: introduction, type of meaning change in revision, main annotation steps and sample of annotation interface. The introduction section provides a brief explanation of the annotation task and the general purpose of the annotation guidelines. The definitions and examples for different revision types are provided in the section type of meaning change in revisions. The annotation steps enlist the scale from the least significant to the most significant changes (i.e. formal change to meaning preserving change to micro-structure change to macro-structure change).

5.2.2 Annotation Process

In addition to the annotation guidelines, the preparation for annotation such as numbering of the sentences according to the sequence and presentation of the revision to ensure that the annotators understand the annotation task. For each paper, there are two annotators: an original author of the paper (labelled as A1) and the other is not an author of the original paper (labelled as A2). Each annotator is presented with the differences between two versions of a paper as revised by different authors in the form of output of *diff* (reviewed in Section 2.4.2). A sample of this is also presented in the Annotation Interface section in the annotation guidelines. For instance, paper 1 has four *diff* output files, thus, each of the annotators is provided with four *diff* outputs of the paper (each of the *diff* output is labelled as d_{v_o,v_r}) (Table 5.1). For each of the *diff* output, revisions are scoped at sentence level, presenting revised sentence pairs to the annotators. Each of these sentences is numbered according to the sequence (i.e. Rev1, Rev2, etc.).

The academic papers used in the annotation process are published in computational linguistics peer reviewed publication, hence, authors are regarded as experts in their field. As this is a specialized field, the non-authors are selected among graduate students and researchers in the field of computational linguistics. All of the annotators are supplied with the annotation guidelines in Appendix C. The annotators are required to read the guidelines prior to annotating each of the revised sentence pairs. For each of the revision, the annotator is required to annotate one of the four types of revision: formal, meaning preserving, micro- and macro-structure changes. Based on these annotations, the inter-annotator reliability measurements (reviewed in Section 2.5.1) are calculated and analysed in Table 5.3.

In order to verify the coherence of annotators towards both our proposed annotation guidelines and the annotation task, the annotators were asked a few questions after they had annotated the drafts of academic papers. The list of questions is presented in Table 5.2). The questions include if they understand the annotation task and is the task difficult. From presentation point of view, the annotators are also asked how accurate are the revisions presented and whether they considered text beyond the revised sentence pairs. These questions can help us to improve the annotation guidelines.

Based on Table 5.2, all of the annotators understood the annotation guidelines. Despite that, majority of the annotators found the task of annotating revision types difficult. This can be partially attributed to the *diff* presentation, as half of the annotators did not find it correct. This gives room for future experimentation on the presentation of revision to authors and annotators. An important indicator from the extended questions was in most revision cases, all annotators did not consider beyond the sentence scope to determine the revision type (Table 5.2). This might indicate that the annotators naturally consider revision types at sentence level.

TABLE 5.2: Qualitative Questions for Human Annotation of Significant Revision Identification

Questions	Yes	No	Unsure
1. Can you understand the annotation guideline?	6	-	-
2. Overall, is the annotation task hard?	5	1	-
3. Most of the time, do you need to consider beyond the sentence scope?	-	6	-
4. Do you think the *diff* was correct?	3	3	-

5.3 Inter-annotator Reliability for Human Annotation of Revision Types

Inter-annotator (or inter-rater or inter-coder) agreement or reliability measurement refers to the degree of agreement between annotators, which can be used to indicate the validity of the coding scheme (Artstein and Poesio, 2008). Widely used inter-rater reliability measurements for computational linguistics namely, simple agreement, Scott π, Cohen κ and Krippendorff α coefficient values (reviewed in Section 2.5.1) are calculated and shown in Table 5.3.

TABLE 5.3: Inter-Annotators Reliability Measurement for Revision Type Categorisation for Drafts of Academic Papers

Paper ID	P1	P2	P3
N Authors	2	4	3
N Annotators	2	2	2
N Set Back-to-back versions	4	4	1
N Revised Sentences	95	128	132
N Agreement	64	97	91
N Disagreement	31	31	41
Simple Agreement (%)	67.4	**75.8**	68.9
Scott π	0.506	0.528	0.539
Cohen κ	0.515	0.534	0.542
Krippendorff α			
Nominal	0.508	0.530	0.539
Ordinal	**0.745**	**0.722**	0.680

Simple agreement is higher compared to π and κ coefficient values which indicates moderate inter-annotator agreement (Table 5.3). De Swert (2012) stated that to consider accepting a variable if $\alpha > 0.67$. The $\alpha_{ordinal}$ values obtained for the drafts of academic papers are greater than 0.67, which make *the revision types coding scheme acceptable*. These annotators have never performed the annotation task before and rely on the annotation guidelines provided. Obtaining an acceptable α value provides an indication of although annotators found the task difficult (Table 5.2), they seem to

TABLE 5.4: Revision Types Distribution for Corpus II as annotated by Human Annotators

Revision Type	Paper 1		Paper 2		Paper 3	
	Annotator		Annotator		Annotator	
	1	2	1	2	1	2
Formal	21	13	11	4	20	19
Meaning Preserving	10	27	16	17	66	56
Micro-structure	11	12	24	13	31	43
Macro-structure	53	43	77	94	15	14

agree on the revision types. α values greater 0.67 might also indicate the clarity of the annotation guidelines which all of the annotators agreed that they *understood the annotation guidelines* (Table 5.2).

Further observation on Krippendorff α values, $\alpha_{nominal}$ is still moderate but $\alpha_{ordinal}$ is higher or substantial. Having higher $\alpha_{ordinal}$ compared to $\alpha_{nominal}$ implies that *it is easier for participants to judge the revision types as a rank ordering* rather than the revision types as individual categories. This corresponds to the feedback obtained from Corpus I, where the impact of change is as followed: Formal $<$ Meaning Preserving $<$ Minor Meaning $<$ Major Meaning Changes.

The total number of revisions according to the revision type is calculated. Table 5.4 shows the revision types distribution vary across papers and annotators. In total, annotator 1 annotated 14.7% formal changes, 25.9% meaning preserving change, 18.6% micro-structure changes and 40.8% macro-structure changes while annotator 2 annotated 10.1% formal changes, 28.2% meaning preserving changes, 19.2% micro-structure changes and 42.5% macro-structure changes. This is reflective of the possible revision types that exist between revised text documents.

Based on the annotation, sample revised sentences are extracted from the drafts of academic papers and presented in Table 5.5. These sentences show the different kinds of sentences for different types of revisions.

5.4 Baselines

This section proposes baseline approaches that consider superficial text differences, as would be identified by most word processors with a "track changes" feature (Iversen, Jan, 2018; *Track changes*). The track changes feature presents edits such as addition and deletion to the readers, where the underlying detection of edits is based on Levenshtein's edit distance (refer to Section 2.3.1.1 for details of Levenshtein's edit distance). However, edit distance alone does not indicate the revision types. Edit distances between the revised sentences can be measured either at word-level (or differences between the words) or character-level (or differences between the characters)

TABLE 5.5: Sample Revision Sentences from Corpus II

Revision Types	**Sample Revision**
Formal (grammar correction)	Some research has investigated syntactic properties of MWEs, to detect their compositionality. Some research has investigated *the* syntactic properties of MWEs, to detect their compositionality.
Meaning Preserving	However, their assumptions were not general for every language, e.g. they assume that the number of a specific type of MWE (light verb constructions) in Persian is much more than the number of the same MWE types in English. However, their assumptions were not easily generalisable across languages , e.g., they assume that the relative frequency of a specific type of MWE (light verb constructions) in Persian is much greater than in English.
Minor Meaning Change (deletion of information that cannot be derived)	Although methods using a bilingual *corpus seem to be more general,* they *still* have a number of drawbacks. Although methods using a bilingual *corpora are intuitively appealing*, they have a number of drawbacks.
Major Meaning Change (information that cannot be derived)	Our best results were *as well as (sometimes much better than) previous studies based on* vector-based approaches. Our best results were *found to be competitive with state-of-the-art results using* vector-based approaches, *and were also shown to complement state-of-the-art methods*.

(for explanation on Levenshtein edit distance refer to section 2.3.1.1). Based on the assumption that different edit distance thresholds lead to different revision types, using a small amount of annotated data to set the thresholds, two baseline approaches are proposed. In the event that a minor edit is made between the revised sentences, the assumption is that it can be categorised as a minor change and if the edit distance is higher, then it is a major change.

5.4.1 Correlation between Human Annotation and Levenshtein's Distance at Word And Character Level

The assumption here is a lot of edits (i.e. a lot of changes such as addition, deletion and modification made) will lead to more significant changes. In order to validate the assumption, the LvD at word- and character-level (reviewed in Section 2.3.1.1) are calculated for each of the revised sentence pairs extracted from the drafts of academic papers and are plotted against the revision types as annotated by the annotators. The Pearson correlation coefficient, r (Section 2.3.4) and the coefficient of determination, r^2 (square of correlation coefficient) between LvD and the revision types as annotated by the annotators are calculated and present in Table 5.6. Table 5.6 shows that LvD and the revision types have weak to moderate correlation. As explained by Maloney (2003), the coefficient of determination measures the amount of variation that can be explained by the r value, where r^2 is the proportion of the variance that is shared by both variables. In our case, r^2 is the percentage *variation for the revision types that can be explained by variations of the Levenshtein's distance values, which is between 10% to 26%* (Table 5.6). Hence, it might not be the case that many edits will result to significant revision. However, this variation is between 10% to 26%, LvD might actually be useful for other revision types. Therefore, we proposed baseline approaches based on LvD at character- and word-level which will be explained in Section 5.4.2.

TABLE 5.6: Correlation between Levenshtein's Distance and Revision Types

Paper	Pearson's Coefficient, r		Co-efficient of Determination, r^2	
	LvDWord	LvDChar	LvDWord	LvDChar
Paper1	0.500	0.512	0.250	0.262
Paper2	0.312	0.335	0.097	0.112
Paper3	0.456	0.512	0.208	0.262

5.4.2 Baseline Methods

For our proposed baseline approaches, the revised sentence pairs, $(s_o, s_r)_k$ are extracted from versioned texts (refer to the preprocessing phase in Figure 4.3 for details). For each of the sentence pairs, the Levenshtein distance (LvD) are calculated as

the number of edits (insertions, deletions, or substitutions) needed to convert from s_o to s_r at either word- (LvDWord) or character-level (LvDChar) (explanation on Levenshtein's edit distance at word- and character-level is available in Section 2.3.1.1). The Levenshtein edit distance value alone cannot directly indicate the revision type but by setting the edit distance range for each of the revision types using annotated data, the range can be used to predict the revision type given any revision sentence pair (the proposed baseline algorithms are provided in Table 5.7.

TABLE 5.7: Range settings algorithms for each paper

Input	Revised Sentence Pairs (S_o, S_r) and count for each annotated revision type, N_i
Output	Levenshtein's Distance (LvD) Range according to Revision Type $(L-U)_i$
Algorithm	Baseline - Levenshtein's Distance Range Set
1:	For each (S_o,S_r)
2:	Generate Levenstein's Distance between (S_o, S_r)
3:	For each revision type, $i \in \{F, MP, Mi, Ma\}$
4:	Calculate the average LvD, $\bar{x}_i = \frac{1}{n_i}\Sigma LvD_i$,
5:	For each revision type, i
6:	Calculate the cut-off between the revision type,
7:	$U_i = \frac{x_i - x_{i-1}}{2} = L_{i+1}$

The aim here is to set the edit distance range for each of the revision types based on annotated data (Table 5.7). First, the average LvD for that revision type is calculated by summing up the total LvD values for that revision type and dividing by the number of revisions for that revision type. For example P1, A1 annotated 21 formal changes, where total LvD value for the 21 revisions is 91, giving an average of 4.33. Then, the average distance is rounded up to the closest integer, which is 5 for the example earlier. For the boundary values between the revision types, for instance, between formal and meaning preserving revision types, the upper and lower bounds for the side-by-side revision types are based on the differences between the average for both revision types and divided by two, to be added up to the earlier average as the cut off value. Using the same example earlier, the average for meaning preserving revision type is rounded up to 9. (Average meaning preserving revision type − average formal revision type) / 2 = (9 − 4) / 2 = 2.5. Round up (4 + 2.5) = 7. Therefore the range of LvD for formal revision is ≤ 7. The same process is repeated for the rest of the revision types. Both LvD for word- and character-level use the same algorithm to set the range although LvDWord is calculated based on word while LvDChar is calculated based on character. The algorithm to set the range for each paper is available in Table 5.7. The LvD range derived for the different revision types from the separate papers at word-level are shown in Table 5.8, while at character-level are shown in Table 5.9.

Even though the calculated range for P1, P2 and P3 are shown in Table 5.8 and

TABLE 5.8: LvDWord Range for Paper 1, Paper 2 and Paper 3

Paper	Revision Type	Round (Average)	Cut-off	Upper Bound	Range
1	F	4	2.5	6.5	<=7
	MP	9	5	14	8-14
	Mi	19	2	21	15-21
	Ma	23		>=22	
2	F	4	2.5	6.5	<=7
	MP	9	6	15	8-15
	Mi	21	2	23	16-23
	Ma	25		>=24	
3	F	2	6	8	<=8
	MP	14	1.5	15.5	9-16
	Mi	17	4	21	17-21
	Ma	25		>=22	

TABLE 5.9: LvDChar Range for Paper 1, Paper 2 and Paper 3

Paper	Revision Type	Round (Average)	Cut-off	Upper Bound	Range
1	F	27	12.5	40	<=40
	MP	52	23	65	41-65
	Mi	98	20	111	66-111
	Ma	138		>=112	
2	F	21	13.5	34.5	<=35
	MP	48	33	81	36 - 81
	Mi	114	17	131	82 - 131
	Ma	148		>=132	
3	F	8	32	40	<=40
	MP	72	12.5	84.5	41-85
	Mi	97	23.5	120.5	86-121
	Ma	144		>=122	

Table 5.9, for evaluation purpose, the LvD range for P1 uses the range set from annotated data from P2, while P2 uses the range derived from P3 and P3 uses the range derived for P1. This is to avoid using the same paper to make the evaluations. In order to compare with our proposed SigRevId computational framework, the experimental setup for our proposed baseline approaches are available in Section 6.3.

5.5 Chapter Summary

We proposed a new task of significant revision identification (SigRevId) but there is neither an existing corpus nor annotation guidelines for the task. This chapter showed

the development of a suitable corpus for significant revision identification. Annotation guidelines to annotate significant revisions between revised sentences are prepared based on the lesson learnt from previous human annotation of software requirement specifications. Based on the human annotation of drafts of academic papers, Krippendorff's $\alpha_{ordinal}$ values obtained are greater than 0.67, which makes *the revision types coding scheme acceptable*.

Other than observing the inter-rater agreement, Levenstein distance (as commonly used in track changes feature in text processor) and revision types annotated by the human, have a weak to moderate correlation, specifically, the variation for Levenstein distance is about 10-20% variation of the revision types. This chapter also proposed a baseline comparison which is based on Levenshtein distance at character- and word-level, similar to the track changes feature in the current text processors. In the next chapter, this corpus will be used for evaluation of our proposed significant revision identification framework.

Chapter 6

A Case Study of Significant Revision Identification

The task of significant revision identification (SigRevId) is defined as classification of revised sentence pairs $(s_o, s_r)_k$ that are extracted from the original text, v_o and revised text, v_r, into one of the four revision types: formal, meaning preserving, micro-structure and macro-structure changes, where macro-structure changes correspond to major meaning change. The core concept of our proposed conceptual framework is the outcome assessment of bi-directional textual entailment between revised sentence pairs to distinguish the different revision types (Figure 3.1). In Chapter 4, the proposed conceptual framework was translated to computational framework as shown in Figure 4.2. In order to demonstrate the applicability of our proposed computational framework, this chapter presents a case study of significant revision identification on a corpus that was introduced in Chapter 5, which consists of drafts of academic papers.

Direct comparison of SigRev approaches to other approaches can not be made if the inputs vary between the approaches and the output cannot be directly compared to human annotated data. Thus, a general process flow for revision type categorisation is proposed, where the inputs are constants and the output of the revision types can be directly compared to the annotated data by humans (Figure 6.1). The output generated is compared against human annotated data of revision types (refer to Section 5.3). This will produce results that can be directly comparable to evaluate the revision type categorisation approaches.

A detailed description of the experimental setup for this case study is provided in Section 6.2. In the previous chapter, two baseline approaches had been proposed which consider superficial string differences, specifically using Levenshtein's edit distance that is similar to the track changes feature in current word processors that presents edits to readers. These baseline approaches are used to compare with our proposed computational framework. This chapter presents the analysis of the results before discussing the implications of our findings.

6.1 Revision Type Categorisation General Process Flow

In order for the results of different categorisation approaches to be comparable, a general process flow to categorise revision types is proposed (Figure 6.1). The flow consists of three main processes: versioned texts pre-processing, revision type categorisation and comparison with human annotators. Versioned texts pre-processing inputs two texts which are versions of one another, and then pre-processes these texts to produce versioned sentence pairs (details of this process in Section 6.2.1). Revision type categorisation inputs the versioned sentence pairs from the previous process. In this phase, different approach to categorise the sentences is allowed. The output of revision type categorisation process is the revision types for the revised sentence pairs. These revision types can be directly compare with annotated data by humans (see Section 6.2.3 for the details). The revised sentence pairs and the annotated data are constant, thus, the results of the revision types can be directly compared to evaluate the performance of revision type categorisation approaches.

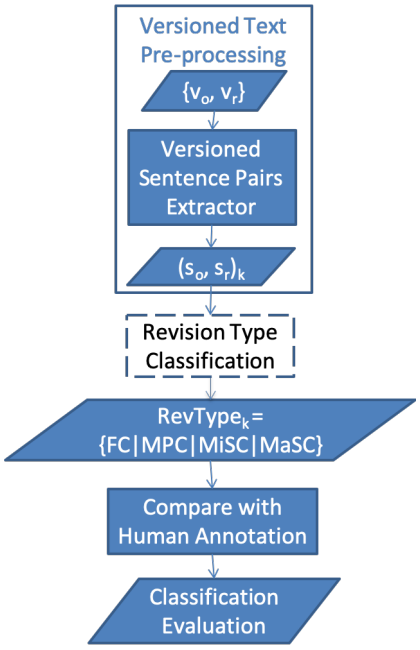


FIGURE 6.1: Revision Type Categorisation General Process Flow

6.2 Significant Revision Identification Experimental Setup

The detailed implementation for each of the phases of SigRev is described in Chapter 4, while this section focuses on our evaluation setup (Figure 6.2). The inputs to our experimental setup are original and revised drafts of academic papers and the output are classification results of the revision type produced by an automated method are compared to human annotation.

The purpose of the experimental setup (Figure 6.2) is to investigate the effect of different entailment decision algorithms on SigRev classification of revision types. The

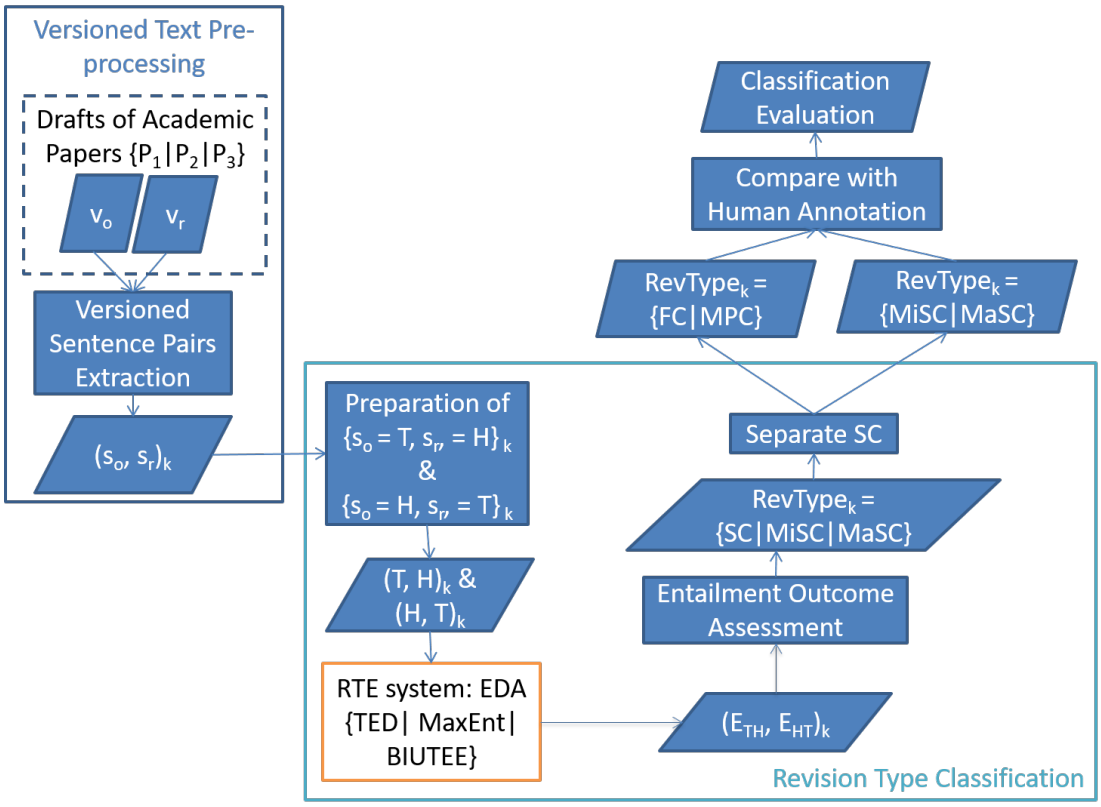


FIGURE 6.2: Experimental setup that consists of three main phases to investigate different entailment decision algorithms (presented using red box) on classification of revision type

summary for the experimental setup for each of the phases are provided in the sub-sections below.

6.2.1 Versioned Texts Pre-processing

The versioned text pre-processing phase (detailed description in Section 4.2) inputs two drafts of an academic paper by different authors, (v_o, v_r) and the outputs are versioned sentence pairs extracted from (v_o, v_r), the sentence pair of k, $(s_o, s_r)_k$. These sentence pairs are converted to text, T and hypothesis text, H pairs, where T = s_o and pair with H = s_r or T = s_r and pair with H = s_o. The pairs of T and H serve as input to the recognition of textual entailment phase.

6.2.2 Recognition of Textual Entailment

The experimental setup for this phase inputs pairs of T and H which are the revised sentence pairs. An existing RTE system; the Excitement Open Platform (EOP) (Magnini et al., 2014), is employed in this phase. The outcome of this phase is textual entailment as recognised by the RTE system, $(E_{TH}, E_{HT})_k$ where E is either entailment or non entailment.

For experimentation purposes, three entailment decision algorithms (EDA) are tested: tree edit distance (TED), classification (MaxEnt) classifier and transformation

based (BIUTEE) (Figure 6.2). In the case of classification based entailment decision algorithm, three different sets of features are tested:

1. the most basic model with bag-of-words (BoW),
2. additional syntactic and semantic dependencies using dictionaries, and
3. additional syntactic, semantic dependencies using dictionaries, named entities and linguistic information.

The brief description of our methods for each of the approaches are listed below (detailed description for each of the entailment decision algorithms is provided in Section 2.6.2):

SigRevTED This variant recognises textual entailment of the revised sentence pair using tree edit distance (TED) (Kouylekov and Magnini, 2005). For T = s_o and H = s_r, T and H are parsed to individual dependency trees. If the edit distance (i.e. the cost of the editing operations such as insertion, deletion and modification) to transform from dependency tree T to dependency tree H is less than a given threshold empirically estimated from the training data, then assign an entailment relation between the revised sentences. The process is repeated to assess the revised sentence pair in the opposite direction or T = s_r and H = s_o.

SigRevMaxEnt This classification based entailment decision algorithm learn a classification model using a maximum entropy (MaxEnt) classifier to combine the outcomes of several scoring functions. The experimental method is based on MaxEnt with bag-of-words (BoW) features, plus similarity scoring and lemmas with the scoring from the RTE system (Magnini et al., 2014). The classifier learns from existing examples of entailment and non-entailment to infer a model that determines if s_o entails s_r or not. The analysis is repeated to determine whether s_r entails s_o.

SigRevMaxEntWNVO This approach is similar to SigRevMaxEnt but adds lexical knowledge to recognise textual entailment of revised sentence pairs. MaxEntWNVO considers hypernym, synonym, part holonym from WordNet (WN) and verb relation of stronger than, can result in and similar from VerbOcean (VO) as features (Wang and Neumann, 2007).

SigRevMaxEntAll This approach is similar to SigRevMaxEntWNVO but this variant uses additional features: part-of-speech (PoS) and dependency relation or tree skeleton (Wang and Neumann, 2007) to recognise textual entailment of revised sentence pairs.

SigRevBIUTEE Bar Ilan University Textual Entailment Engine (BIUTEE) (Stern and Dagan, 2014) parses T and H to separate parsed trees, similar to TED. However, the difference with BIUTEE is it considers more than just insertion, deletion and

modification. In BIUTEE, a sequence of transformations is performed to transform parsed tree T to parsed tree H, either preserving fully or partially the meaning of the original sentence, before deciding if T and H entail. The variant uses transformation based EDA to recognise the textual entailment of revised sentence pairs. In SigRevBIUTEE, evaluation is done for T = s_o and H = s_r, and the process is repeated to evaluate for T = s_r to H = s_o.

6.2.3 Classification of Revision Type

This phase is crucial as it implements our proposed bi-directional entailment assessment concept (Table 3.1). The setup here is to assess $(s_o \models s_r, s_r \models s_o)$ = {(true, true), (true, non entail), (non entailment, non entail)} according to the rules presented in Table 3.1, where the outcome of the evaluation can be surface change, micro-structure change (MiSC) or macro-structure change (MaSC). Additional process is implemented to differentiate formal change (FC) and meaning preserving change (MPC) in surface change (details of the implementation are available in Section 4.4.2).

The classification results for each of the RTE approaches are compared against human annotation of the corpus presented in Chapter 5. When an approach selects the same revision type as an annotator for a revised sentence pair, this is labelled as true positive, TP. However, when an approach selects a revision type as 'positive or correct' but the annotator labelled it as 'negative or incorrect' revision type, this revision is labelled as false positive, FP. In the case, when an approach selects a revision type as 'negative or incorrect' but the annotator labelled it as 'positive or correct' revision type, this revision is labelled as false negative, FN. When an approach selects a revised sentence pair as 'not that category' and is categorised by annotator as 'not that category', that revised sentence pair is a true negative (TN) (detailed explanation of TP, FP, FN and TN values can be found in Section 2.5). The evaluation measurements are calculated (refer to the explanation on the evaluation measurement Section 2.5) for each of the approaches. There are two annotators, who sometimes disagree. Thus, we compared the classification results separately and presented the results separately for the two annotators. The results are compared and analysed to understand significant revision identification with regards to the RTE approaches. The results and analysis are presented in Section 6.4.

6.2.4 A Revised Sentence Pair Example for Significant Revision Identification

In order to demonstrate the workings of our experimental setup (Figure 6.2), an example of a revised sentence pair is used as input to the textual entailment module with tree edit distance (TED) as the choice of entailment decision algorithm (EDA):

s_o: `In this project, we use the translations of MWEs and their components to estimate the semantic similarity between them.`

s_r: `In this research, we use the translations of MWEs and their components to estimate the relative degree of compositionality of the MWE.`

For the example above, using the tree edit distance based EDA, the textual entailment outcome is *Entail* for $s_o \Rightarrow s_r$ and *True* for $s_r \Rightarrow s_o$. When the results are into the SigRevId classification module, the revision category for (s_o, s_r) is detected as surface change (SC).

For sentence pairs that are detected as SC, these sentence pairs are inputted into an additional module named differentiation between formal change (FC) and meaning preserving change (MPC) (the explanation of this module is available in Section 4.4.2). In this module, string similarity measurement is used to measure the similarity between the sentences within the sentence pair. The assumption is that sentences within the sentence pair that have higher string similarity are most likely to be spelling and grammar corrections or formatting (i.e. formal change) while bi-directionally entailed sentences that are less similar will most likely be a re-phrase or meaning preserving change.

6.3 Baseline Experimental Setup

In addition to comparing different types of entailment decision algorithm (EDA), results using SigRev approaches are compared to our two proposed baseline approaches based on string edits (Section 5.4.2). The experimental framework for the baseline approaches is shown in Figure 6.3. The inputs to the experimental setup for baseline approach are revised sentence pairs extracted from v_o and v_r, $(s_o, s_r)_k$, making the results directly comparable.

6.4 Revision Type Classification Results and Analysis

The generated classification results are compared against human annotated values (Section 5.3) and are evaluated using multi-class classification evaluation methods: recall, precision and F_1-measure (refer to Table 2.1 for precision, recall and F_1 formulas) (Sokolova and Lapalme, 2009). This section presents micro- and macro-averaged revision classification results for the overall revision types, while subsequent sections present the classification performance based on the individual revision type. Table 6.1 and 6.2 show the micro- and macro-averaged precision, recall and F_1-scores to summarise how the different approaches perform across the drafts of academic papers against the annotated data by different annotators.

SigRev based approaches perform better than approaches that are based on Levenshtein edit distance (LvD) for significant revision identification (Table 6.1 and 6.2). **Generally, evaluating entailment between the revised sentence pair can assist in significant revision identification compared to approaches that are based solely on LvD**. SigRev approaches depend on the entailment decision algorithms to recognise

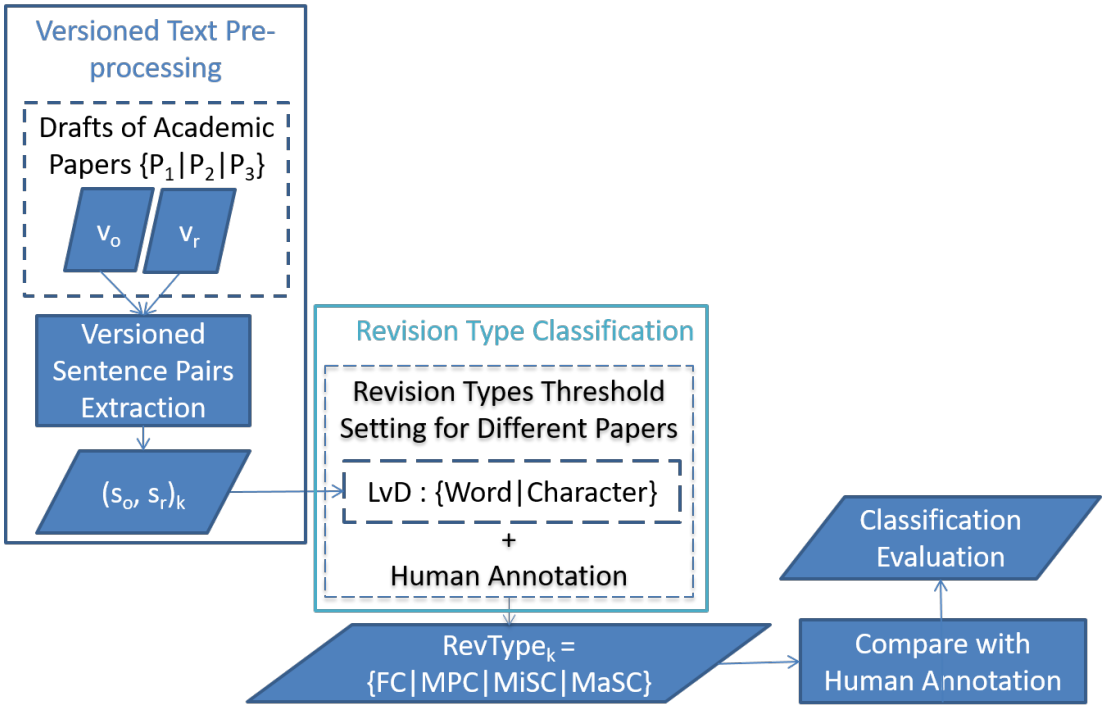


FIGURE 6.3: Baseline approaches experimental setup

TABLE 6.1: Significant revision identification results against annotation by annotator 1, A1 for micro- and macro-averaged Precision, Recall and F_1-score

Approach	Micro-averaged			Macro-averaged		
	Precision	Recall	F_1	Precision	Recall	F_1
SigRevTED	**.541**	**.541**	**.541**	.498	**.502**	**.500**
SigRevMaxEnt	.513	.513	.513	**.507**	.492	**.500**
SigRevMaxEntWNVO	.518	.518	.518	.500	.493	.496
SigRevMaxEntAll	.448	.448	.448	.460	.352	.398
SigRevBIUTEE	.476	.476	.476	.443	.449	.446
LvDWord	.448	.448	.448	.424	.484	.452
LvDChar	.442	.442	.442	.426	.483	.453

the entailment within the sentences in the revised sentence pairs. In the subsections below, detailed analysis of the performance of the SigRev variants are presented.

6.4.1 Tree Edit Distance

Overall, SigRevTED performed best compared to all of the methods compared (Table 6.1 and 6.2). The underlying approach to recognising textual entailment is tree edit distance (Kouylekov and Magnini, 2005) (review is available in Section 2.6.2.1). The approach goes beyond inserting, deleting or substituting nodes from dependency tree of s_o to the dependency tree of s_r. The sequence of transformations considers the dependency structure of the source label. Generally when revising a sentence, we

TABLE 6.2: Significant revision identification results against annotation by annotator 2, A2 for micro- and macro-averaged Precision, Recall and F_1-score

Approach	Micro-averaged			Macro-averaged		
	Precision	Recall	F_1	Precision	Recall	F_1
SigRevTED	**.566**	**.566**	**.566**	.501	.537	.518
SigRevMaxEnt	.527	.527	.527	.499	.515	.507
SigRevMaxEntWNVO	.549	.549	.549	**.530**	**.541**	**.535**
SigRevMaxEntAll	.482	.482	.482	.490	.379	.427
SigRevBIUTEE	.499	.499	.499	.463	.493	.477
LvDWord	.394	.394	.394	.372	.450	.407
LvDChar	.397	.397	.397	.394	.455	.422

will consider the original structure of the sentence too. In addition, each transformation has its cost, with insertion of a word based on the inverse document frequency. Hence, if a word appears more frequently in the training set such as stopword (i.e a word that is too frequent that it might not be useful), the cost of insertion becomes 0 while inserting a word that hardly appears will have more weight. However, deletion cost is 0, placing more emphasis on insertion. On the other hand, substitution cost relies on the similarity generated from a dependency based thesaurus; words that have no similarity will have a similarity value of 0, therefore the cost of substitution is the same as inserting the word. This gives an advantage to categorising either meaning preserving changes (as some substitution of words are indeed similar words) or macro-structure changes, where there is no similarity at all. The overall cost of the transformation considers the edit cost from dependency tree s_o to dependency tree s_r and the edit cost to insert all words into s_r. For this directional approach, s_o entails s_r if the sequence of transformations is below the threshold that separates positive and negative training sets. The training sets are supplied in the RTE system. SigRevTED method also evaluates if s_r entails s_o.

SigRev assesses the bi-directional textual entailment outcome between s_o and s_r to categorise the revision type. The assessment depends on the accuracy of the textual entailment and in the case of SigRevTED, TED is able to recognise the textual entailment of revised sentence pairs quite accurately. When the revised sentence pairs that have been correctly categorised by SigRevTED are analysed (Table 6.3), we find that the method is suitable for sentence pairs that are syntactically similar and have high lexical overlap with minor edits and are mostly preserving the meaning. An example is selected to reflect the strength of SigRevTED in contrast to the other of the methods:

s_o = `Therefore, we excluded languages with coverage of less than half of the dataset size.`

s_r = `Therefore, we excluded languages with MWE translation coverage of less than 50%.`

For this example, all other methods failed to categorise this revised sentence pair correctly, but SigRevTED was successfully applied to classify this pair as a meaning

preserving (MP) change. The difference between the revised sentence pair is insertion of "MWE translation" which is insertion of noun to an existing noun (i.e "coverage"), while substitution of "half of the dataset size" with "50%". The insertion produced higher cost because it is not a stop word but a substitution which produced almost 0 cost due the similarity bringing the total cost low. If total cost is below the cost set through the training set, hence s_o entails s_r at both directions or meaning preserving change. When the entailment outcomes between that revised sentence pair are analysed for SigRevMaxEntWNVO and SigRevMaxEntAll, these approaches produced no entailment for both textual entailment directions. SigRevMaxEnt and SigRevBIUTEE produce $s_o \leftarrow s_r$ as True but do not detect entailment in the other direction, thus failing to categorise the sentence pair correctly.

SigRevTED works too for major meaning change when two sentences of a pair have limited lexical (word-level) overlap. During revision of a paper, revision might not necessarily produce a full sentence. Despite this, generally, SigRev approaches are able to categorise sentence pairs with incomplete sentences or different length but with high lexical overlap and syntactically similar. An instance where all methods apart from SigRevTED fail to detect any entailment between the sentences is:

s_o = `In vanilla LDA, the number of topics is fixed, whereas in`

s_r = `In standard LDA, the user needs to figure out how to appropriately set the number of topics` T.

Most methods have difficulty in categorising sentence pairs that are very different syntactically or with limited word overlap which either can be meaning preserving or micro-structure revision. This can be observed from the confusion matrix (Table 6.3), which displays the amount of wrongly categorised instances of meaning preserving and micro-structure changes. SigRevTED is based on parse trees and the comparison of edit operations to transform from one tree to another, hence, having sentence pairs with no similarity will result in no entailment. The example sentence pair below is annotated as a micro-structure change but SigRevTED could not recognise any entailment between the sentences:

s_o = `For research purposes, Twitter provides access to two feeds, which represent a 1% and 5 representative sample of the total feed.`

s_r = `By default, the streaming API provides access to a 1% sample of the total Twitter feed, and this can be increased to 5% for research purposes.`

In summary, significant revision identification using SigRevTED performed best not only because of bi-directional entailment evaluation, the entailment decision algorithm parsed the revised sentences to trees and considered edit operations between the parsed trees which is helpful in revision type identification.

TABLE 6.3: Confusion Matrix for SigRevTED

		SigRevTED			
		FC	MPC	MiSC	MaSC
Annotator 1	FC	35	8	4	4
	MPC	16	32	15	30
	MiSC	5	17	14	30
	MaSC	2	4	28	111
Annotator 2	FC	26	4	4	2
	MPC	21	33	16	30
	MiSC	9	20	19	20
	MaSC	2	4	22	123

6.4.2 Different Feature Sets in Classification Based Entailment Decision Algorithms

Although Maximum Entropy (MaxEnt) classifier is used for classification based entailment decision algorithm (EDA), three different sets of features are experimented with (as summarised in Section 6.2). Out of the three sets of features, SigRevMaxEntWNVO performed best (Table 6.1 and 6.2). In the case of SigRevWNVO, other than bag-of-words (BoW) and lexemes with similarity scores (only features used in SigRevMaxEnt), additional lexical features which include hypernym, synonym, part of holoymn from WordNet (WN), verb relations of stronger then, can result in and similar from VerbOcean (VO), performed better than just using BoW and lexemes. This shows that **the kind of feature set used for classification based on EDA does make a difference in significant revision identification**.

For observation of possible ways to apply linguistics as features for classification based EDA, an example of sentence pairs with significant length was chosen where the author rated as MaSC, while non-author rated as MiSC:

s_o = `In Table 2, 3 and 4, we show how often each language was selected as the top 10 languages using LCS`

s_r = `In Table 2, 3 and 4, we show how often each language was selected in the top-10 languages over the combined 100 (10x10) folds of nested 10-fold cross validation, based on LCS`

SigRevMaxEnt produced $s_o \Rightarrow s_r$ as false while $s_r \Rightarrow s_o$ as true, similar to SigRevBIUTEE. However, for the rest of the methods, the revised sentences were found to entail at both directions. For this particular sentence pair, information was added to the revised sentence, increasing the sentence length significantly. The consideration of synonym, hypernym, part of holonym and other verb relations as features did not make much difference to the entailment outcome. Nevertheless, one possible explanation why there was not much categorisation performance difference between SigRevMaxEnt and SigRevMaxEntWNVO is that specialised terms are used

in drafts of academic papers. For instance, `100 (10x10) folds of nested 10-fold cross validation`, the individual words are available in WN and VO dictionaries. However `cross validation` is used as adjacent terms which bring specialised meaning in machine learning are available as separate words in WN and VO. Another observation made for this particular example, for SigRevMaxEntAll, using the additional linguistic information as features is not helpful, instead when this linguistic information is applied to parse trees as in the SigRevBIUTEE approach, this information is relevant for distinguishing entailment and non entailment between the sentences. There are a variety of ways to apply linguistic information, however how this information can be integrated into RTE methods to improve revision sentences classification requires thorough study.

Compared to the other two classification based EDA, SigRevMaxEntAll performed worst. This is also reflected in the confusion matrices (Table 6.4), where the number of revision sentence pairs correctly categorised as FC, MPC and MiSC (or TP) for SigRevMaxEntAll are lower compared to the other methods. SigRevMaxEntAll has the highest true positive values for macro-structure change. This suggests that when using all the features, the representation is not distinctive enough to distinguish entailed sentences. This could be due to noise, either some of the features might not be useful or the sparsity due to increased dimensionality of the representation. Using all of the features, SigRevMaxEntAll categorises most revisions as MaSC to the point of over generalising as MaSC. This can be observed for instances that are categorised correctly using SigRevMaxEntWNVO and SigRevMaxEnt and are also categorised correctly by SigRevMaxEntAll.

Nonetheless, there are a few exceptions that SigRevMaxEntAll successfully categorised these sentence pairs, for example:

s_o = `In topic modelling, words (observed data) are seen as evoked by latent topics (unobserved) that exist in a document.`

s_r = `In topic modelling, words are considered to be evoked by latent topics in a document.`

This particular example, SigRevMaxEntAll and SigRevTED are able to recognise that the sentence pair is entailed bi-directionally because in such case, dependency and PoS information are helpful.

6.4.3 Knowledge-based Transformations

BIUTEE has been shown to have higher accuracy compared to the other entailment decision algorithms for other datasets and tasks (Magnini et al., 2014), however, for significant revision identification, the performance is average compared to the other EDAs (Table 6.1 and 6.2). Even when compared with the confusion matrices of other SigRev approaches (Table 6.3 and 6.4), the number of correctly categorised revision sentence pairs (TP) by SigRevBIUTEE is also less (Table 6.5). The issue faced when using a knowledge-based method such as BIUTEE might be similar to the issue faced

TABLE 6.4: Confusion Matrices for SigRevMaxEnt, SigRevMaxEntWNVO and SigRevMaxEntAll with cells filled with blue colour are true positives as compared to annotator 1 and 2 for the respective revision types: formal change (FC), meaning preserving change (MPC), micro-structure change (MiSC) and macro-structure change (MaSC)

		SigRevMaxEnt			
		FC	MPC	MiSC	MaSC
Annotator 1	FC	34	6	7	4
	MPC	15	11	30	37
	MiSC	5	3	29	29
	MaSC	2	1	34	108
Annotator 2	FC	25	3	6	2
	MPC	21	12	34	33
	MiSC	8	5	31	24
	MaSC	2	1	29	119
		SigRevMaxEntWNVO			
		FC	MPC	MiSC	MaSC
Annotator 1	FC	36	7	4	4
	MPC	17	14	24	38
	MiSC	5	6	23	32
	MaSC	2	0	32	111
Annotator 2	FC	28	3	3	2
	MPC	20	18	26	36
	MiSC	10	6	27	25
	MaSC	2	0	27	122
		SigRevMaxEntAll			
		FC	MPC	MiSC	MaSC
Annotator 1	FC	11	2	7	31
	MPC	7	12	28	46
	MiSC	3	5	15	43
	MaSC	1	0	23	121
Annotator 2	FC	8	0	4	24
	MPC	10	15	28	47
	MiSC	2	4	20	42
	MaSC	2	0	21	128

when using dictionaries such as WN and VO: the available relevant information might be limited for the dataset we are using.

There are some exceptions where only SigRevBIUTEE is able to recognise the correct textual entailment. For the example below, other methods consider the sentence pair as bi-directional entailed but SigRevBIUTEE recognises it as one way only entailment:

s_o = `for each document, a new distribution of mixture components` G_m `is sampled from a base distribution` G_0 `. Both of these distributions are`

TABLE 6.5: Confusion Matrix for SigRevBIUTEE

		SigRevBIUTEE			
		F	MP	Mi	Ma
Annotator 1	F	33	6	8	4
	MP	12	5	32	44
	Mi	4	4	23	35
	Ma	1	1	35	108
Annotator 2	F	26	3	4	3
	MP	15	7	32	46
	Mi	8	6	28	26
	Ma	1	0	34	116

distributed according to a Dirichlet Process (DP), and are controlled by parameter γ and α_0 respectively. The generative story of a word using the HDP can be summarised as follows. (1) Choose a base distribution $G_0 \sim DP(\gamma, H)$; (2) for each document m, generate distribution $G_m \sim DP(\alpha_0, G_0)$; (3) draw a latent topic z from the document's mixture component distribution G_m; (4) draw a word from the chosen topic z.

s_r = The particular implementation of non-parametric topic model we experiment with is Hierarchical Dirichlet Process (HDP: where, for each document, a distribution of mixture components G_m is sampled from a base distribution G_0 as follows. (1) Choose a base distribution $G_0 \sim DP(\gamma, H)$; (2) for each document m, generate distribution $G_m \sim DP(\alpha_0, G_0)$; (3) draw a latent topic z from the document's mixture component distribution G_m; (4) draw a word from the chosen topic z.

For this particular example, although the SigRevBIUTEE is able to produce the correct entailment outcome, when the sentences are analysed, the points (i.e. (1) - (4)) at the end of s_o and s_r are the same, which can be ignored as BIUTEE extends from TED (Stern et al., 2012). The differences are s_o has two extra sentences and the term "new" but these differences are summarised in s_r in a sentence. BIUTEE incorporates knowledge-based transformations (entailment rules) with a set of predefined tree-edits other than insert, delete and substitute In addition, BIUTEE has a more effective way to determine if two texts entail. Hence, the correct entailment is attributed to the knowledge-base for various relations of words such as synonym, to transform from the parsed tree for s_o to the parsed tree of s_r and vice versa. This example also shows error can occur in the sentence scoping approach, which will be discussed in detail in Section 6.10. Wrong scoping can complicate the process of SigRevID.

6.4.4 Levenshtein's Edit Distance based Approaches

LvD based approaches have the highest number of true positive (TP) for formal change, meaning that they correctly identify formal change revision sentence pair as formal change (Table 6.6), although LvD based approaches also wrongly categorise high number of revision sentence pairs that are not formal change as formal change or false positive for formal change compared to the other approaches (Table 6.3, 6.4 and 6.5). LvD approaches have high recall values but low precision because this approach relies on lexical differences between the revised sentence pairs, which for actual FC is effective. However, for most of the revised sentence pairs, the sentences are similar lexically with slight differences which change the meaning entirely. This explains why LvD approaches suffer from a huge amount of false positive. The analysis of LvD performance for each of the revision types is provided in the sections below. LvD based approaches performed worse compared to bi-directional entailment evaluation based approaches (SigRev).

TABLE 6.6: Confusion Matrices for Levenshtein's Word and Character Level

		LvDWord				LvDChar			
		FC	MPC	MiSC	MaSC	FC	MPC	MiSC	MaSC
Annotator 1	FC	47	3	0	1	45	3	2	1
	MPC	36	28	13	16	40	20	19	14
	MiSC	13	15	16	22	12	9	24	21
	MaSC	19	25	33	68	15	23	39	68
Annotator 2	FC	33	2	0	1	31	2	2	1
	MPC	41	28	17	14	42	23	21	14
	MiSC	18	17	10	23	20	8	19	21
	MaSC	23	24	35	69	19	22	42	68

LvD based approaches require setting a distance threshold for each revision type (refer to Section 5.4 for the algorithms to set the range) based on annotated data. A separate annotated set is required for determining the best threshold values. We used different drafts of academic paper to set the threshold. This could be the cause of the methods being less effective.

6.5 Surface Change: Distinguishing Formal and Meaning Preserving Changes

The previous section presents micro- and macro-averaged results for all four of the revision types. This section presents the individual revision type categorisation performance for formal and meaning preserving revision types in comparison against

annotator 1 (Table 6.7) and for annotator 2 (Table 6.8), which are calculated based on the confusion matrices generated (Table 6.3, 6.4, 6.5 and 6.6).

TABLE 6.7: Performance for formal (FC) and meaning preserving (MPC) changes against annotation by annotator 1, A1 for Precision, Recall and F_1-score

Annotator 1	FC			MPC		
	Precision	Recall	F_1-score	Precision	Recall	F_1-score
SigRevTED	.603	.686	.642	.525	**.344**	**.416**
SigRevMaxEnt	.607	.667	.636	.524	.118	.193
SigRevMaxEntWNVO	.600	.706	.649	.519	.151	.233
SigRevMaxEntAll	.500	.216	.301	**.632**	.129	.214
SigRevBIUTEE	**.660**	.647	**.653**	.313	.054	.092
LvDWord	.409	**.922**	.566	.394	.301	.341
LvDChar	.402	.882	.552	.364	.215	.270

TABLE 6.8: Performance for formal (FC) and meaning preserving (MPC) changes against annotation by annotator 2, A2 for Precision, Recall and F_1-score

Annotator 2	FC			MPC		
	Precision	Recall	F_1-score	Precision	Recall	F_1-score
SigRevTED	.448	.722	.553	.541	**.330**	**.410**
SigRevMaxEnt	.446	.694	.543	.571	.120	.198
SigRevMaxEntWNVO	.467	.778	.583	.667	.180	.283
SigRevMaxEntAll	.364	.222	.276	**.789**	.150	.252
SigRevBIUTEE	**.520**	.722	**.605**	.438	.070	.121
LvDWord	.287	**.917**	.437	.394	.280	.327
LvDChar	.277	.861	.419	.418	.230	.297

For formal change categorisation, a majority of SigRev approaches have higher F_1-score compared to LvD approaches with SigRevBIUTEE having the highest precision and F_1-score while SigRevMaxEntAll has the lowest recall and F_1-score against both annotators (Table 6.7 and 6.8). When the sentences that fall under the formal change category are analysed, the sentences have more syntactic corrections and formatting. BIUTEE incorporates both linguistics and world knowledge and this information is able to support categorisation of formal change. SigRev approaches utilise bi-directional textual entailment first to filter out changes with no meaning change, followed by lexical matching (i.e. Jaro-Winkler string similarity) to distinguish formal changes and meaning preserving changes (our proposed approach as presented in Section 4.4.2).

LvD approaches have the highest recall for formal changes (Table 6.7 and 6.8). Higher recall shows that the selected items for LvD is relevant. This can be likely attributed to LvD based approaches using annotated data to set the range. As demonstrated in Table 5.6, between 10% to 26% variation for the revision types that can be explained by variations of the Levenshtein's edit distances, and formal changes are

very likely to fall into this range. SigRevMaxEntAll performed worst at categorising formal changes. SigRevMaxEntAll simply didn't predict as many formal changes. One possible explanation is that using all of the features might not be distinctive in recognising the entailment between revised sentence pairs that have revisions due to errors in spelling and grammar or even formatting. Hypothetically, these revisions can be filtered out first using spelling and grammar checkers.

SigRevTED performs best at categorising meaning preserving change (Table 6.7 and 6.8), and has the highest TP values for meaning preserving changes compared to the other methods (Table 6.3, 6.4, 6.5, 6.6). The strengths and weaknesses of SigRevTED approach are presented in the earlier section (Section 6.4.1). However, when the confusion matrix (Table 6.3) is studied, there are two potential problems identified in detecting meaning preserving change: the similarity threshold can be wrong (i.e. actual MP changes that fall into formal changes or FN is 16) and the EDA may fail to detect bi-directionally entailed revised sentence pairs (i.e. actual MP changes that fall into micro- and macro-structure changes (FN): 15 and 30 respectively). When comparing the confusion matrices (Table 6.4 and 6.5), all SigRev approaches displayed the same problem. When these sentence pairs are analysed, generally, a few problems are identified in MP change categorisation. Current EDAs are not strong enough to recognise bi-directional entailment of revisions such as

- change from active to passive voice
- re-phrasing that has limited lexical overlap
- substitution of words that might not be linked to an existing dictionary

As shown in the example earlier, revision might not necessarily be one full sentence and instead could be a partial sentence or may involve multiple sentences. Some examples of meaning preserving changes that are not detected by any of the methods tested include the following:

Case of complex re-phrase of a sentence

s_o = `Panlex and Google Translate are more appropriate for our task.`

s_r = `This leaves translation resources such as Panlex and Google Translate.`

Case of words are not linked in dictionary 'method' cannot be linked to 'experiments'.

s_o = `We evaluate our method over two datasets, as described below.`

s_r = `We evaluate our experiments with two datasets which are described below:`

Case of switching from passive to active voice

s_o = `The measures are designed and normalized in a way that we get the score`

s_r = In each case, we normalize the output value to the range different strings.

Due to the complexity of meaning preserving revisions, as future work, other paraphrase approaches may be considered to filter meaning preserving changes. Consideration could also be given to improve identification of bi-directionally entailed revised sentence pairs through creation of additional training examples.

6.6 Micro-structure Change Categorisation

Although SigRevMaxEnt performed best for categorisation of micro-structure change, *all methods are weak at detecting micro-structure changes* (with micro-averaged F_1-score < 0.4) as shown in Table 6.9 and 6.10. For our proposed conceptual framework (Figure 3.1), revised sentence pairs that are entailed one way only, regardless of the entailment directions, are categorised as micro-structure change. As presented in the confusion matrices (Table 6.3, 6.4 and 6.5), a majority of micro-structure changes (as annotated by human) are categorised wrongly as macro-structure changes. Most of the entailment decision algorithms categorise revised sentence pairs that are micro-structure change as having no entailment at all regardless of the direction, resulting in SigRev approaches categorising the revision pairs as macro-structure changes. The interaction between directionality of entailment and EDA performance would be interesting to study in more depth.

TABLE 6.9: Performance for micro-structure change (MiSC) against annotation by annotator 1, A1 for Precision, Recall and F_1-score

Approach	MiSC		
	Precision	Recall	F_1-score
SigRevTED	.230	.212	.220
SigRevMaxEnt	**.290**	**.439**	**.349**
SigRevMaxEntWNVO	.277	.348	.309
SigRevMaxEntAll	.205	.227	.216
SigRevBIUTEE	.235	.348	.281
LvDWord	.258	.242	.250
LvDChar	.286	.364	.320

SigRevMaxEnt performed best for micro-structure changes, showing that deriving similarity of words and lexemes from the training set is useful. MaxEnt classifier recognises textual entailment by learning the similarity scoring function using the features BoW and lexemes (Wang and Neumann, 2007). The revised sentence pairs that are annotated as micro-structure changes but fail to be correctly categorised by any of the approaches are analysed. There are a small number of cases where annotators annotated this as micro-structure change but for most cases of revision, annotators annotated adding a single sentence as a macro-structure change. An example of such

TABLE 6.10: Performance for micro-structure change (MiSC) against annotation by annotator 2, A2 for Precision, Recall and F_1-score

Approach	MiSC		
	Precision	Recall	F1-score
SigRevTED	.311	.279	.295
SigRevMaxEnt	.310	**.456**	**.369**
SigRevMaxEntWNVO	**.325**	.397	.358
SigRevMaxEntAll	.274	.294	.284
SigRevBIUTEE	.286	.412	.337
LvDWord	.161	.147	.154
LvDChar	.226	.279	.250

case is when a sub-heading is deleted which combine that sub-section with the earlier section. Currently in our proposed computational framework (Figure 4.2), when a single sentence is added or deleted, the sentence is paired with null sentence, which directly results in no entailment at all and is therefore categorised as macro-structure change. This is a weakness in the current approach and as future work, we might consider a better approach to handle such cases.

Other cases where an error occurred include revised sentence pairs that are syntactically and lexically very similar but have minor edits which add or delete information. For instance this revised sentence pair is annotated by annotator 1 as micro-structure while annotator 2 annotated as macro-structure change but computational approaches categorise as formal change. The sentence pairs are the same except the difference between s_o and s_r is substitution of "consistently outperforms" to "performs very similarly to" as highlighted. High lexical overlap is the main cause all methods fail.

s_o = `In Figure 3, we see that over the SVM,` $Skew_AM$ `and Maxent with a byte-bigram tokenization, the classifier trained over WikiTweets data consistently outperforms the classifier trained over Wikipedia data.`

s_r = `In Figure 3, we see that over the SVM,` $Skew_AM$ `and MaxEnt with a byte-bigram tokenization, the classifier trained over WikiTweets data performs very similarly to the classifier trained over Wikipedia data.`

6.7 Macro-structure Change as Significant Revision

SigRev approaches performed better than Levenshtein edit distance (LvD) based approaches at categorising macro-structure revision as presented by the F_1-score in Table 6.11 and 6.12. LvD based approaches calculate edit distances and this shows that a larger number of surface edits do not necessary imply major meaning change. The distinction of the SigRev approaches lie with the assessment of the textual entailment between revised sentence pairs. Rather than relying on the edits alone, **our proposed significant revision identification framework provides an alternative approach to**

identify macro-structure change or major meaning change between revised sentence pairs which assess the entailment.

TABLE 6.11: Performance for macro-structure change (MaSC) against annotation by annotator 1, A1 for Precision, Recall and F_1-score

Approach	MaSC		
	Precision	Recall	F_1-score
SigRevTED	.634	.766	**.694**
SigRevMaxEnt	.607	.745	.669
SigRevMaxEntWNVO	.600	.766	.673
SigRevMaxEntAll	.502	**.834**	.627
SigRevBIUTEE	.565	.745	.643
LvDWord	.636	.469	.540
LvDChar	**.654**	.469	.546

TABLE 6.12: Performance for macro-structure change (MaSC) against annotation by annotator 2, A2 for Precision, Recall and F_1-score

Approach	MaSC		
	Precision	Recall	F_1-score
SigRevTED	**.703**	.815	**.755**
SigRevMaxEnt	.669	.788	.723
SigRevMaxEntWNVO	.659	.808	.726
SigRevMaxEntAll	.531	**.848**	.653
SigRevBIUTEE	.607	.768	.678
LvDWord	.645	.457	.535
LvDChar	.653	.450	.533

For categorisation of macro-structure change, SigRevTED performs best (Table 6.11 and 6.12), but SigRevMaxEntAll has the highest recall value (Table 6.4). Unlike categorising the other three types of revision (i.e. formal, meaning preserving and micro-structure changes), if a proposed method is able to categorise some of the revised sentence pairs correctly (TP) for macro-structure change but the same approach categorises actual revised sentence pairs that are macro-structure changes as no meaning change or minor meaning change (FN), this will lead to readers missing out on important revisions. Likewise, if many non-significant revisions are categorised as significant revisions (i.e. a high number of false positives) this defeats the purpose of our objective to identify significant revisions. *The argument here is that missing out on significant revision is much worse than non-significant revision being categorised as significant because the author would be unaware of revisions with meaning change performed by other authors.* The target is a high amount of true positives, and a low amount of false positives. Thus, F_1-score evaluation is a better choice for macro-structure change, where it considers an average of recall and precision (refer to Table 2.1 for precision, recall and F_1-score formulas).

Based on our analysis on the strength and weaknesses of SigRev approaches, most approaches are able to detect macro-structure changes. One of the characteristics of macro-structure revised sentence pairs is sentences being added or deleted or sentences that have limited lexical and syntactical overlap, for example:

s_o = `Topics learnt are interpreted as senses induced by the model.`

s_r = `For both vanilla LDA and HDP, the sense assignment for a given instance is determined by simply returning the sense with the highest probability.`

From the confusion matrices (Table 6.3, 6.4, 6.5 and 6.6), most false negatives for our proposed textual entailment assessment approaches fall into micro-structure changes while for LvD approaches, the false negatives occur across a full range of revision types (i.e. formal, meaning preserving and micro-structure changes). This indicates that **our proposed approach to assess the textual entailment of the revised sentence pairs are better at categorising meaning changing revisions when compared to LvD based approaches, which are based only on syntactic edit distances**.

Categorising macro-structure changes faces similar problems in categorising micro-structure revisions with sentence pairs that are syntactical and lexically similar but with seemingly minor edits and that in the case of macro-structure changes, changes the meaning entirely.

6.8 Surface change vs Text-Base change

This section discusses the observation and analysis on revised sentence pairs that are correctly selected as surface and text-base changes while other pairs fail to be categorised as meaning and no meaning changes. Surface change (or no meaning change) and text-base change (meaning change) are the upper level of Faigley and Witte's (1981) taxonomy (Figure 2.2). For the purpose of differentiating between surface change (SC) and text-base change (TBC), our proposed framework remained the same but the textual entailment outcome for micro- and macro-structure changes were collapsed as TBC. Similarly for the human annotation of drafts of academic papers, formal and meaning preserving changes were collapsed as surface changes while micro- and macro-structure changes were collapsed as text-base changes. The precision, recall and F_1-score for categorisation of surface and text-based changes are calculated and presented in Table 6.13.

Based on F_1-score (Table 6.13), the *results are comparative for LvDChar, LvDWord and SigRevTED for surface change categorisation*. Our proposed LvDChar approach (refer Section 5.4) is based on the edit distance of the characters between original and revised sentences, while using annotated data to determine the edit distance threshold for the revision types. Nevertheless, LvDChar, LvDWord and SigRevTED are all based on edit distance. Spelling and grammar corrections, formatting and re-phrasing, all fall under surface change. This shows **edit distance based approaches seem to perform better at categorising surface change or revision with no meaning change**.

TABLE 6.13: Surface and text-base revision types Precision, Recall and F_1-score categorisation results comparing between SigRevTED, SigRevMaxEnt, SigRevMaxEntVOWN, SigRevMaxEntAll, SigRevBIUTEE, LvDWord and LvDChar

Approach	Surface Change			Text-base Change		
	Precision	Recall	F_1	Precision	Recall	F_1
SigRevTED	.765	.632	.692	.775	.867	.819
SigRevMaxEnt	**.857**	.458	.597	.719	.948	.818
SigRevMaxEntVOWN	.851	.514	.641	.739	.938	**.827**
SigRevMaxEntAll	.780	.222	.346	.643	**.957**	.769
SigRevBIUTEE	.849	.389	.533	.696	.953	.804
LvDWord	.613	*.792*	.691	*.822*	.659	.732
LvDChar	.647	.75	**.695**	.809	.720	.762

For significant revision identification, revision sentence pair, s_o and s_r that are bi-directional entailed have surface change or no meaning change which can be either formal or meaning preserving change, while revised sentence pairs with one-way entailment only or no entailment at all have text-base changes or revisions with meaning change. A bi-directional textual entailment approach is used to detect sentences that are paraphrased (Androutsopoulos and Malakasiotis, 2010; Zhao and Wang, 2010). SigRev approaches obtained low recall values for surface change or low precision for text-base change categorisation (Table 6.13). The possible reason why SigRev approaches detected a high number of either one way entailment or no entailment at all although the RTE approaches should have recognised entailment in both directions, are due to the *variation in meaning preserving change revised sentence pairs*. The revised sentence pairs that fall into such error in categorisation can be divided into sub-categories, namely:

Difference in length between original and revised sentences

s_o = `Moreover, some types of MWE`

s_r = `Moreover, some MWEs whether they are compositional or non-compositional.`

s_o = `They consider non-compositional MWEs to be those candidates aligned to the same target unit.`

s_r = `They consider non-compositional MWEs to be those candidates that align to the same target language unit, without decomposition into word alignments.`

Extensive rephrase

s_o = `Most recent studies focus on semantic properties of MWEs; they measure the semantic similarity of the MWEs with their components using different sources , such as Wordnet, and techniques, such`

as or distributional similarity relative to a corpus (e.g. Latent Semantic Analysis (LSA)

s_r = Much of the recent work on MWEs focuses on their semantic properties, measuring the semantic similarity between the MWE and its components using different resources , such as , and techniques, such as WordNet or distributional similarity relative to a corpus (e.g. based on Latent Semantic Analysis

s_o = And finally, the experiments are done on English versus non-English languages (mostly European) The assumption behind these proposed methods might not be applicable for all languages.

s_r = And finally, most experiments have been carried out on English paired with other European languages, and it is not clear whether the results translate across to other language pairs.

For SC categorisation, SigRevMaxEntAll is an outlier where low recall value is obtained. This can be due to the entailment decision algorithm (EDA) based on maximum entropy (MaxEnt) classification using all of the features (i.e. hypernym, synonym, part of holoymn from WordNet (WN), verb relations of stronger then, can result in and similar from VerbOcean (VO), word dependency, dependency with part-of-speech (PoS) and tree skeleton). Changes to part-of-speech (PoS) and dependency tuples have been shown to correlate to edit importance or ratings of surface and text-base changes (Goyal et al., 2017). In order to be categorised as surface change, the sentences within the revised sentence pair must entail in both directions. However, the SigRevMaxEntAll approach, using all the features, failed to recognise the sentences entailing in both directions. Rather, the revised sentences were recognised as one-way entailment only or no entailment at all especially for revisions with minimal edits such as adding a word. For an example of a sentence pair where other approaches categorised them correctly except for SigRevMaxEntAll, this particular sentence pair had minimal edit but were combined for rephrasing purpose:

s_o = Manual analysis of the false positives reveals that they are sometimes due to the inclusion of few English terms in an otherwise non-english message. But are generally due to the same sequence of letters as an English word meaning something in a non-English language.

s_r = Manual analysis of the false positives reveals that they are sometimes due to the inclusion of few English terms in an otherwise non-english message , but are generally due to the same sequence of letters as an English word meaning something in a non-English language.

Generally, *SigRev approaches performed better at categorising text-base change* with higher F_1-score (Table 6.13). Although LvD approaches have higher precision values and SigRev approaches have higher recall values, the number of actual revision sentence pairs that is identified as text-base change (i.e. true positive) is higher for SigRev approaches.

SigRev approaches performed better at categorising revision with meaning change compared to LvD approaches. When LvD approaches are compared with SigRev approaches, LvD approaches identify more revised sentences that are actually text-base change as surface change, or sentence pairs with smaller edit distances are actually not surface changes. SigRevMaxEntVOWN performed best at categorising text-base change. SigRevVOWN uses features such as structural information, hypernym, synonym, part of holoymn from WordNet (WN), verb relations of stronger then, can result in and similar from VerbOcean (VO). This indicates that such features are able to recognise entailment correctly, categorising revised sentence pairs with meaning change correctly. Hypothetically, using features from dictionaries should be able to identify meaning change. By comparison, our proposed **bi-directional entailment evaluation is a better option to categorise text-base change or revision with meaning change**.

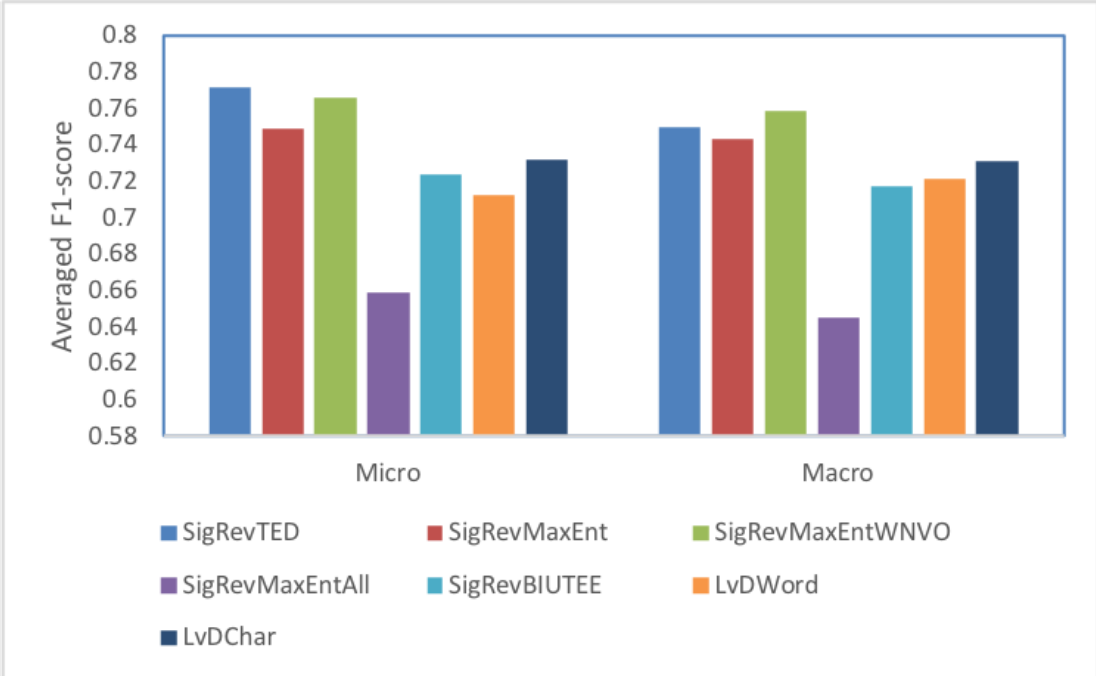


FIGURE 6.4: Micro- and macro averaged F_1-score for the overall surface and text-based changes categorisation results for Annotator 1

Table 6.13 presents the categorisation results for the separate categories (i.e. surface and text-base changes). The micro- and macro-averaged precision, recall and F_1-score are calculated for the overall categorisation results and micro- and macro-averaged F_1-score are presented in Figure 6.4. Overall, SigRevTED performed best. Other than bi-directional textual entailment evaluation, SigRevTED uses entailment decision algorithm that is based on tree edit distance. This form of combinations is able to categorise revisions according to meaning change better.

6.9 Other Observed Revisions and Entailment Decision Algorithm

Other than the main four revision categories, our proposed conceptual model (Figure 3.1) consists of other revisions as observed through introspective analysis such as deletion of duplicate pronoun which falls under formal change and restatement under meaning preserving change. Although our proposed computational approach, SigRev does not explicitly detects these revisions separately, the strategies employed by these entailment decision algorithms are related to certain observed revisions. For example, deletion of redundant pronoun and subject verb agreement correction, the strategy in tree edit distance (TED) EDA is to transform the original and revised sentences to dependency trees and compare the trees. This strategy considers pronoun, subject, and verb. This is reflected in the classification results for formal change using TED EDA. Another example is the observed revision of adding new information by adding a new sentence. All the EDA strategies are able to detect this and is reflected with higher F_1-score for all of the approaches in detecting macro-structure change compared to classification results of the other revision types. Indirectly, rather than individually detecting these revisions, our proposed assessment of textual entailment for revised sentences considers these revision types due to strategies employed in the EDAs. The summary of the observed revisions in our proposed conceptual model in relation to the best performing EDA for each of the category is presented in Table 6.14.

6.10 Limitations of Recognition of Textual Entailment System

Based on the analysis of our results in the earlier sections, our proposed conceptual framework based on bi-directional entailment of evaluation of revised sentence pairs (Figure 3.1) is valid, but the current recognition of textual entailment (RTE) tools are not able to adequately recognise the textual entailment of the revised sentence pairs due to the great variations in revisions. One of the most obvious cases is sentence pairs that are lexically and syntactically similar but with minor edits that can possibly result in any of the four types of revisions: formal, meaning preserving, micro-structure or macro-structure change, which makes revision type categorisation a challenging task.

When revising a text, revisions might not be necessarily result to forming one full sentence. Based on our sentence pair analysis, bi-directional entailment evaluation can help to address revision within partial sentences, giving advantage to our proposed conceptual framework compared to LvD approaches.

Conceptually, approaches such as edit distance and transformation based EDA should be similar to how humans manually revise, which is transforming from the original to a revised form. Our empirical results (Section 6.4 and 6.7) only show that SigRevTED perform well but not SigRevBIUTEE. Here the reason is speculated to be similar to the problem faced by SigRevMaxEntWNVO, where the dictionary entries could be limited.

TABLE 6.14: Different kinds of revision changes in relation to the strategy used in entailment decision algorithm

Meaning Change Category	Revision Observation	Entailment Decision Algorithm - Strategy
Formal	• delete redundant pronoun • subject verb agreement correction	BIUTEE - parse tree, edit operations
Meaning Preserving	• restatement within round brackets or parentheses • similar word or phrase substitution	TED - dependency tree, insertion
Micro-structure	• add extra information to existing sentence such as adding a Noun Phrase, description, adjective • confirming what is not	MaxEnt - Bag-of-Words, similarity measure, lemmas
Macro-structure	• add new information (add new sentence(s))	TED - dependency tree, insertion

When SigRev approaches are contrasted with Levenshtein edit distance (LvD) approaches, LvD calculation itself does not require annotated data but setting thresholds for mapping of the revision types requires annotated data and a strategy to determine good thresholds. Our empirical results for macro-structure changes categorisation (Section 6.7) show that edit distance based approaches are weak.

There is no RTE system that caters for the task of significant revision identification (SigRevId), hence having no specific training set for the RTE system. The Excitement Open Platform (EOP) for recognition of textual entailment (RTE) system (Magnini et al., 2014) can be used for different tasks such as information retrieval (IR), information extraction (IE), question answer (QA) and summarisation (SUM). When revised sentence pairs are compared with the four tasks (i.e. IR, IE, QA and SUM) that use RTE, the revised sentence pairs are quite similar to the task of IE, as the revised sentences are not question answer, although there is the possibility that some revised sentences are summaries of longer sentences or some terms from the original sentence might still exists in the revised sentence. However due to the approaches used for RTE, a training set is required. Thus, currently for the training set in the RTE system, the task for the revision sentence pairs is set to IE (Note: No modification is done to the training set). in future work, detail study will be conducted to determine whether specific training sets are required for the RTE system for the task of significant revision identification.

From our error analysis, there are cases of macro-structure changes where all approaches fail to categorise the sentence pairs correctly. A particular example of such a sentence pair is provided as below. When we examine this, we might suggest that this pair should have been annotated as micro-structure change instead. Hence, the distinction between micro- and macro-structure changes can be better conveyed to human annotators.

s_o = `The other interface for accessing Twitter messages is the streaming API, which provides a real-time feed of a subset of messages submitted to Twitter.`

s_r = `The other interface for accessing Twitter messages is via the Streaming API, which provides a real-time feed of all messages submitted to Twitter.`

6.11 Chapter Summary

This chapter presents a detailed analysis of significant revision identification based on a case study using a corpus of drafts of academic papers. We compare our proposed approach based on bi-directional entailment with edit distance based approaches.

Our findings indicate that

- Our proposed framework is able to effectively categorise revision type based on meaning change, through consideration of both possible directions of entailment between the revised sentence pairs, $(s_o \Rightarrow s_r, s_o \Rightarrow s_r)$. Our proposed approach provides a strategy for identification of significant revisions that abstracts away from small edits that have less impact on text interpretation.

- Our empirical results show that a tree edit distance based entailment decision algorithm (EDA) used for recognition of textual entailment (RTE) performed best for significant revision identification. Tree edit distance based EDA converts sentences to dependency trees and calculates the edit costs required to transform one parse tree into another. Having a training set for revision types categorisation can potentially help to improve recognition of textual entailment for revised sentence pairs.

- There are quite a few revised sentence pairs that are categorised incorrectly. Theoretically, this is not due to the proposed approach based on entailment not being fully able to capture the complexity of the task. Rather, through analysis of the sentences, it is the failure of the specific existing recognition of textual entailment methods in recognising the true entailment, due to range of variation in revision sentence pairs. Revision is a complex process.

- Variations in correspondences between revised sentence pairs pose a great challenge to significant revision identification. Empirical results supported this, with different entailment decision algorithms performing best for different sentence

pairs reflecting different revision types. Adding more linguistic components to computational processing might not necessarily improve the identification of significant revision such as SigRevMaxEntAll.

Chapter 7

Conclusion, Contributions and Future Work

This research started with a broad question: How do we identify significant revisions, given two versions of a text, without needing to read them from the beginning to the end? Previous studies and current word processors cannot fully support such capabilities. This thesis investigated the question in-depth and based on this question, three research questions were derived as below:

- What are the different kinds of revision changes to be considered as significant revision for revised text documents in a multi-author environment?
- Given two versions of a text document in a multi-author environment, how do we identify significant revisions?
- How do we evaluate the task of significant revision identification?

This research is built upon a taxonomy for analysing revisions (Faigley and Witte, 1981). We proposed a new task to identify significant revisions in a multi-author environment. The task of significant revision identification is defined as given two versions of a text, identify the revised sentence pairs as one of the four revision types: formal, meaning preserving, micro- and macro-structure changes. The evaluation measurements from multi-class classification task were adopted for the task of significant revision identification. Two cases of versioned texts were used, namely, software requirements specification using use case specification and collaborative scientific article writing. The versions of use case specification were used for analysis purpose, while the drafts of the scientific articles were developed to evaluate the task of significant revision identification. From this thesis, the concept of bi-directional textual entailment assessment of revised sentence pairs to identify significant revisions have been demonstrated to be computationally feasible. Our proposed computational framework is compared against edit distance based approaches, which is similar to the current track changes capability built in most word processors as the baseline approaches. Other than edits, significant revision identification depends on the linguistic information used in the recognition of textual entailment system such as part-of-speech tags, hypernym, synonym and verbs. This chapter provides a summary of our investigation (Section 7.1) and the contributions of our work in Section 7.2. In Section 7.3, the

limitations of our work are discussed and followed by the presentation of possible future work.

The human agreement between authors and non-authors on the four-category meaning change annotation task for drafts of academic papers is moderate agreement (i.e. alpha Krippendorff = 0.745). The categorisation performance of the four-category revision for our proposed approach is micro-averaged F_1 = 0.541, which is set as the baseline for future comparison. Our proposed approach works better better than the baseline models derived using edit distance only for formal and macro-structure changes identification. Bi-directional textual entailment is commonly used for paraphrase detection (see Section 2.6.2), however, our proposed approach performed below average for identification of meaning preserving change.

7.1 Summary of Chapters

For Chapter 3, the different kinds of revision changes are investigated and what is considered as significant revision for revised text documents in a multi-author environment is proposed. Through introspective analysis and human feedback from both the authors and non-authors reviewing the changes, we conformed to the taxonomy (Faigley and Witte, 1981) that revisions can be divided into two primary groups: surface changes (i.e. no meaning change) and text-base changes (i.e. meaning change). Surface change is further divided into formal and meaning preserving changes, while text-base change is divided into micro- and macro-structure changes, with macro-structure change regarded as significant revision, which answered to research question 1. Our proposed conceptual framework extends the taxonomy for analysing revision (Faigley and Witte, 1981).

The works in the domain of psycho-linguistics are examined. In written discourse, meaning is built up from word to phrase and sentence, while sentences entail to create cohesion in the discourse and eventually build up to global meaning. Revisions can result in meaning change within the sentence (i.e. local meaning change) or beyond the sentence (i.e. global meaning change). When an original sentence, s_o is revised, producing a revised sentence, s_r, if there is no meaning change, s_o entails s_r denoted as $s_o \Rightarrow s_r$, where a typical human reader would infer the meaning of s_r by reading s_o and $s_r \Rightarrow s_o$, by reading s_r, one can infer the meaning of s_o. For cases of meaning change, likely $s_o \Rightarrow s_r$ is false and vice versa and even if $s_o \Rightarrow s_r$ is true, $s_r \Rightarrow s_o$ might not necessary be true. This concept serves as the basis of our proposed conceptual framework, which is introduced in Chapter 3. Based on this concept of meaning change, the textual entailment directions of a revised sentence pair extracted from two versions of a text document were assessed to differentiate the revision types: bi-directional entailment of the revised sentence pair is surface change or no meaning change, one-way textual entailment is micro-structure change and no entailment at all is significant change. This answered to our second research question.

This conceptual framework was then translated to a computational implementation and is explained in Chapter 4. Given two versions of a text, the significance of the revised sentence pairs were identified using our proposed computational implementation (described in Chapter 4). Recognition of textual entailment system was used to support our computational implementation. We explored three different types of entailment decision algorithms in recognition of textual entailment system to support differentiation of revision changes between revised sentence pairs. This answered our second research question further. We proposed a computational framework (Figure 4.2) to identify significant revisions.

Chapter 5 describes the development of an annotated corpus of drafts of academic papers for the purpose of evaluation of significant revision identification. The annotators consisted of authors and non-authors, who had been provided with an annotation guideline developed based on the lesson learned from the previous feedback. Inter-rater agreement obtained from the annotation was moderate and we maintain that the utility of the corpus we produced can be used generally to evaluate approaches for the task of significant revision identification. Furthermore, this corpus can be extended using the annotation guidelines we had crafted and refined.

Chapter 6 presents the results and analysis for the task of significant revision identification using the evaluation corpus we developed. We demonstrated that assessment on bi-directional textual entailment outcome of revised sentence pairs effectively classified revisions in terms of meaning change. The performance of various entailment decision algorithms (i.e. tree edit distance, classification and transformation based) as the basis for the recognition of textual entailment (RTE) system were tested and the results were compared to two baseline approaches that are based purely on word or character edit distance. Therefore for our third research question, in order to evaluate the task of significant revision identification, an evaluation corpus was developed and tested using our proposed computational framework for significant revision identification and baseline approaches. The evaluation measurements enabled various approaches to be compared. We not only demonstrated that assessing both the entailment directions could improve significant revision identification, different entailment decision algorithms worked for different revision types of sentence pairs.

7.2 Contributions

Our main contribution is bi-directional textual entailment assessment between revised sentence pairs for significant revision identification where paraphrase had been shown to be unable to fully support the detection of the different types of revision. The difference between the taxonomy for analysing revision (Faigley and Witte, 1981) and our proposed conceptual framework is that our framework prescribes a formalised method to distinguish different revision types and the novelty of our approach lies in the way revision types are assessed: assessment of bi-directional textual entailment

outcome of the revised sentence pair. Although the concept of bi-directional entailment of texts has been applied in the task of paraphrase detection (Androutsopoulos and Malakasiotis, 2010; Zhao and Wang, 2010), we adapted it to more than just paraphrase detection, but to classify revision according to meaning change (Table 3.1).

At a higher level, this work contributes to text revision in a multi-author environment. Although there is work that investigated edit importance (Goyal et al., 2017) as perceived by reviewers, to our the best of our knowledge, there is no such capability to identify significant revisions or minor and major meaning changes, given two versions of a text, revised by different authors. Our introspective analysis is a detailed analysis of revisions by multiple authors. We hypothesize that such a capability will be able to assist authors during the revision process in a multi-author environment. We propose a new task called *significant revision identification*. We define the task of significant revision identification as categorising revised sentence pairs to one of the four revision types: formal, meaning preserving, micro- and macro-structure changes. A general process flow for revision type classification (Figure 6.1) is proposed for this task, so that new revision type classification approaches can be directly comparable. Furthermore, we developed annotation guidelines and an evaluation corpus, where any approaches proposed for this task can be used for direct comparison.

Another major contribution is from having no such feature to having a computational approach to significant revision identification. Our third contribution is the design, implementation, and evaluation of significant revision identification between two versions of a text document. We have proposed that RTE approaches can be effectively used to model significant revision. Based on the results and analysis, we are able to provide insights into what works and what does not work for significant revision identification. Our empirical results show that the entailment decision algorithm (EDA) based on tree edit distance, which converts sentences to parse trees and calculates the edit cost to transform from one parse tree to another, overall, performs best at significant revision identification between revised sentence pairs when compared to other approaches. The approach used in transformation based EDA is similar to the tree edit distance EDA, however, instead of just considering the edit costs, transformation based EDA has additional sequences of transformations: words are transformed from the parse tree of the original sentence to the parse tree of the revised sentence. Word or character based edit distance alone has no indication of which revision type it falls under and does not consider the dependency structure of the sentences. Nevertheless, the edit distance based approaches used for comparison in this thesis used annotated data to set the edit distance range for the different revision types. Although all three approaches use edits, how the edits are considered in the approaches influence the revision type categorisation outcome. When comparison is made between tree edit distance, transformation based EDA and edit distance based approaches, we argue that EDA based on tree edit distance is quite similar to how revision is performed by most authors: reading the original sentence and performing edit operations such as inserting, modifying and deleting the original sentence to form the new revised

sentence while keeping a certain dependency structure of the original sentence.

7.3 Limitations and Future Work

A major limitation of our proposed computational solution to significant revision identification is that the proposed approach relies on recognition of textual entailment (RTE) approach. The unsatisfactory revision categorisation result we obtained was not due to the failure of our proposed conceptual framework, but is rather due to the limitation of the entailment decision algorithms that are not yet strong enough to recognise the textual entailment of the revised sentence pair correctly. Furthermore, the current RTE system cannot support the variation of revised sentence pairs. This motivates us to improve the current recognition the textual entailment approach to effectively recognise the textual entailment of the revised sentence pairs for future endeavours.

Another limitation of our proposed approach is that it relied on the existing textual entailment training sets which were task specific, namely information extraction (IE), question-answer (QA), information retrieval (IR) and summarisation (SUM), and none of these was for significant revision identification. We rely on the training or development sets that came together with the RTE system. The IE task was chosen as the closest match to our task of revised sentence categorisation, as other tasks such as question answer (QA) was not relevant in this case while information retrieval (IR) and SUM tasks were not directly applicable to revised sentence pairs. The prediction was that a training set specific for revised sentences where there are instances of the entailment between original and revised sentences and the entailment between the revised and original sentence would improve the performance of revision categorisation. For future directions, either training sets for different tasks are be combined or investigation into which task that is the most suitable for recognising textual entailment for revised sentence pairs. Another possible option is to explore textual entailment systems that use Wikipedia as training set (Zanzotto and Pennacchiotti, 2010). Alternatively, another possibility is to propose an approach to create reliable annotated training sets for recognising the entailment of revised sentence pairs.

Our current framework limits the comparison to two versions of a text at one time. In the case of collaborative writing, although multiple versions can exist, the end goal of the revision is to diverge to common subject matter, for example, multi-bloggers on health issues, researchers publishing scientific articles or a group working on academic written assignment. We argue that if there exists a tool which can pinpoint the author that had made the meaning change, discussion can be initiated to reach an agreement that will not only improve revision experience but strengthen the understanding as a team. It will be interesting if there exists an approach that can detect meaning change between multiple versions at one time. This is one possible future work.

We have presented a computational implementation using a linguistic approach to differentiate revision types according to meaning change. Nevertheless, there are

other existing approaches to semantics such as formal approach (McCready, Yabushita, and Yoshimoto, 2014), statistical analysis (Mozafari, Hashemi, and Hamzeh, 2011) to semantics changes (Wijaya and Yeniterzi, 2011). One possible future direction is to explore the categorisation model built for Wikipedia dataset such as by Yang et al. (2017) for significant revision identification.

Revision of text documents in a collaborative environment is widely practised in a number of fields, namely multi-author blogging, refinement of legislative policies and preparation of teaching materials, which we have not explored yet. Furthermore, the types of articles used in our experimentation are expert authors although the opinions of non-authors are taken into consideration. However, if novice writers were to be explored, different types of revision are required. As future work, we will apply our proposed framework to revised text documents from other fields.

The focus for the annotation task of this research is observe how authors and non-authors rate the revision according to the four-category of meaning change. Thus, the comparison between author and non-author, rather than a more usual approach of using three annotators. Nevertheless, for future work when developing larger dataset, consideration will be given to three annotators, where agreement will be based on the two out of the three annotators.

The task of significant revision identification is formulated as a multi-class classification problem where revision can fall into one of the four-types. Thus, the use of F-metric as evaluation measure. However, for the case of significant revision identification, recall is more important than precision because if significant revisions are categorised as insignificant by an approach, this will cause the authors to miss out on important revisions. For future work, recall will be considered as evaluation metric instead of F-metric.

Our broader aim of identifying significant revision is to assist readers in prioritising which revisions to focus on. Clearly, if we would like to improve revision experience among the authors, there are many aspects of the human side that needs to be examined such as presentation and effectiveness of the revisions identified.

7.4 Closing Remark

On the whole, a new task is explored: significant revision identification between versioned text documents in a multi-author environment, which current word processors are lacking of. The bigger aim is to assist authors prioritise the revisions, especially when transitioning from one draft by one author to another. We maintain the claim that improving revision experience among authors is important. On the whole, we consider two cases of versioned text documents: software requirement specification and collaborative scientific article writing. The task of significant revision identification is challenging and interesting because an effective method to distinguish revision type should be able to accommodate for unforeseen revision sentence pairs when used

by a different corpus. We took the first step towards automatically identifying the edits with meaning change that can improve the revision efficiency in terms of attention and time by authors to concentrate on edits with meaning change. We hope that this will revolutionise the way multi-author documents are revised.

Bibliography

Achananuparp, Palakorn, Xiaohua Hu, and Xiajiong Shen (2008). "The evaluation of sentence similarity measures". In: *International Conference on Data Warehousing and Knowledge Discovery*. Springer, pp. 305–316.

Aditomo, Anindito, Rafael A Calvo, and Peter Reimann (2011). "Collaborative writing: Too much of a good thing? Exploring engineering students' perceptions using the Repertory Grid". In:

Agirre, Eneko et al. (2016). "SemEval-2016 Task 2: Interpretable Semantic Textual Similarity". In: *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. San Diego, California: Association for Computational Linguistics, pp. 512–524. URL: http://www.aclweb.org/anthology/S16-1082.

Akhmatova, Elena and Diego Molla (2006). "Recognizing textual entailment via atomic propositions". In: *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Textual Entailment*. Springer, pp. 385–403.

Aly, Mohamed (2005). "Survey on multiclass classification methods". In: *Neural networks* 19, pp. 1–9.

Androutsopoulos, Ion and Prodromos Malakasiotis (2010). "A survey of paraphrasing and textual entailment methods". In: *Journal of Artificial Intelligence Research* 38, pp. 135–187.

Apache OpenOffice Wiki. *Track changes*. URL: https://wiki.openoffice.org/wiki/Track_changes.

Artstein, Ron and Massimo Poesio (2008). "Inter-coder agreement for computational linguistics". In: *Computational Linguistics* 34.4, pp. 555–596.

Attig, Anja and Petra Perner (2011). "The Problem of Normalization and a Normalized Similarity Measure by Online Data." In: *Tran. CBR* 4.1, pp. 3–17.

Baecker, Ronald M. et al. (1993). "The User-centered Iterative Design of Collaborative Writing Software". In: *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*. CHI '93. Amsterdam, The Netherlands: ACM, pp. 399–405. ISBN: 0-89791-575-5. DOI: 10.1145/169059.169312. URL: http://doi.acm.org.ezp.lib.unimelb.edu.au/10.1145/169059.169312.

"Chapter 11 - Groupware and Computer-Supported Cooperative Work" (1995). In: *Readings in HumanâĂŞComputer Interaction*. Ed. by RONALD M. Baecker et al. Interactive Technologies. Morgan Kaufmann, pp. 741 –753. ISBN: 978-0-08-051574-8. DOI: https://doi.org/10.1016/B978-0-08-051574-8.50077-7. URL: http://www.sciencedirect.com/science/article/pii/B9780080515748500777.

Barzilay, Regina and Noemie Elhadad (2003). "Sentence alignment for monolingual comparable corpora". In: *Proceedings of the 2003 conference on Empirical methods in natural language processing*. Association for Computational Linguistics, pp. 25–32.

Barzilay, Regina and Lillian Lee (2003). "Learning to paraphrase: an unsupervised approach using multiple-sequence alignment". In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, pp. 16–23.

Barzilay, Regina and Kathleen R McKeown (2001). "Extracting paraphrases from a parallel corpus". In: *Proceedings of the 39th annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 50–57.

Benesty, Jacob et al. (2009). "Pearson correlation coefficient". In: *Noise reduction in speech processing*. Springer, pp. 1–4.

Berant, Jonathan et al. (2013). "Semantic Parsing on Freebase from Question-Answer Pairs." In: *EMNLP*. Vol. 2. 5, p. 6.

Bhagat, Rahul and Eduard Hovy (2013). "What is a paraphrase?" In: *Computational Linguistics* 39.3, pp. 463–472.

Biuk-Aghai, Robert P, Christopher Kelen, and Hari Venkatesan (2008). "Visualization of interactions in collaborative writing". In: *2008 2nd IEEE International Conference on Digital Ecosystems and Technologies*. IEEE, pp. 97–102.

Boiarsky, Carolyn (1984). "A model for analyzing revision". In: *Journal of Advanced Composition*, pp. 65–78.

Bonk, Curtis Jay and Kira S. King (1995). "Computer Conferencing and Collaborative Writing Tools: Starting a Dialogue About Student Dialogue". In: *The First International Conference on Computer Support for Collaborative Learning*. CSCL '95. Indiana Univ. Bloomington Indiana, USA: L. Erlbaum Associates Inc., pp. 22–26. ISBN: 0-8058-2243-7. DOI: 10.3115/222020.222045. URL: http://dx.doi.org.ezp.lib.unimelb.edu.au/10.3115/222020.222045.

Boonthum, Chutima (2004). "iSTART: Paraphrase recognition". In: *Proceedings of the ACL 2004 workshop on Student research*. Association for Computational Linguistics, p. 55.

Boriah, Shyam, Varun Chandola, and Vipin Kumar (2008). "Similarity measures for categorical data: A comparative evaluation". In: *red* 30.2, p. 3.

Bos, Johan (2014). "Recognizing textual entailment and computational semantics". In: *Computing meaning*. Springer, pp. 89–105.

Bronner, Amit and Christof Monz (2012). "User edits classification using document revision histories". In: *EACL*. Assoc. for Computational Linguistics, pp. 356–366.

Button, Kathryn, Margaret J Johnson, and Paige Furgerson (1996). "Interactive writing in a primary classroom". In: *The reading teacher* 49.6, pp. 446–454.

Calvo, Rafael A et al. (2011). "Collaborative writing support tools on the cloud". In: *IEEE Transactions on Learning Technologies* 4.1, pp. 88–97.

Cheatham, Michelle and Pascal Hitzler (2013). "String similarity metrics for ontology alignment". In: *International Semantic Web Conference*. Springer, pp. 294–309.

Chklovski, Timothy and Patrick Pantel (2004). "Verbocean: Mining the web for fine-grained semantic verb relations". In: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*.

Clinchant, Stéphane, Cyril Goutte, and Eric Gaussier (2006). "Lexical entailment for information retrieval". In: *European Conference on Information Retrieval*. Springer, pp. 217–228.

Cohen, William, Pradeep Ravikumar, and Stephen Fienberg (2003). "A comparison of string metrics for matching names and records". In: *Kdd workshop on data cleaning and object consolidation*. Vol. 3, pp. 73–78.

Dagan, Ido and Oren Glickman (2004). "Probabilistic textual entailment: Generic applied modeling of language variability". In: *Learning Methods for Text Understanding and Mining*, pp. 26–29.

Dagan, Ido, Oren Glickman, and Bernardo Magnini (2006). "The PASCAL recognising textual entailment challenge". In: *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising tectual entailment*. Springer, pp. 177–190.

Dagan, Ido et al. (2013). "Recognizing textual entailment: Models and applications". In: *Synthesis Lectures on Human Language Technologies* 6.4, pp. 1–220.

Daxenberger, Johannes and Iryna Gurevych (2013). "Automatically Classifying Edit Categories in Wikipedia Revisions." In: *EMNLP*, pp. 578–589.

De Swert, Knut (2012). "Calculating inter-coder reliability in media content analysis using KrippendorffâĂŹs Alpha". In: *Center for Politics and Communication*, pp. 1–15.

Dix, Stephanie (2006). "âĂIJWhat did I change and why did I do it?âĂİ Young writers' revision practices". In: *Literacy* 40.1, pp. 3–10.

Du, Helen S et al. (2016). "Collaborative writing with wikis: an empirical investigation". In: *Online Information Review* 40.3, pp. 380–399.

Dzikovska, Myroslava O, Rodney D Nielsen, and Claudia Leacock (2016). "The joint student response analysis and recognizing textual entailment challenge: making sense of student responses in educational applications". In: *Language Resources and Evaluation* 50.1, pp. 67–93.

Ede, Lisa S and Andrea A Lunsford (1990). *Singular texts/plural authors: Perspectives on collaborative writing*. SIU Press.

Faigley, Lester and Stephen Witte (1981). "Analyzing revision". In: *College composition and communication* 32.4, pp. 400–414.

Fernando, Samuel and Mark Stevenson (2008). "A semantic similarity approach to paraphrase detection". In: *Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics*. Citeseer, pp. 45–52.

Fish, Robert S, Robert E Kraut, and Mary DP Leland (1988). "Quilt: a collaborative tool for cooperative writing". In: *ACM SIGOIS Bulletin*. Vol. 9. 2-3. ACM, pp. 30–37.

Fitzgerald, Jill (1987). "Research on revision in writing". In: *Review of educational research* 57.4, pp. 481–506.

Free Software Foundation, Inc. (2016). *GNU Diffutils*. URL: https://www.gnu.org/software/diffutils/.

Gail, H Richard et al. (2016). *Method and apparatus for automatic detection of spelling errors in one or more documents*. US Patent 9,465,791.

Ghuge, Swapnil and Arindam Bhattacharya (2014). "Survey in Textual Entailment". In: *Center for Indian Language Technology, retrieved on April*.

Giampiccolo, Danilo et al. (2007). "The third pascal recognizing textual entailment challenge". In: *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*. Association for Computational Linguistics, pp. 1–9.

Glickman, Oren (2006). *Applied textual entailment*. Publisher not identified.

Goyal, Tanya et al. (2017). "An Empirical Analysis of Edit Importance between Document Versions". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2780–2784.

Gwet, Kilem L (2014). *Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters*. Advanced Analytics, LLC.

Haake, Jörg M and Brian Wilson (1992). "Supporting collaborative writing of hyperdocuments in SEPIA". In: *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*. ACM, pp. 138–146.

Hadjerrouit, Said (2014). "Wiki as a collaborative writing tool in teacher education: Evaluation and suggestions for effective use". In: *Computers in Human Behavior* 32, pp. 301–312.

Hashemi, Homa B and Christian D Schunn (2014). "A tool for summarizing studentsâĂŹ changes across drafts". In: *International Conference on Intelligent Tutoring Systems*. Springer, pp. 679–682.

Hirao, Tsutomu et al. (2004). "Dependency-based sentence alignment for multiple document summarization". In: *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, p. 446.

Iversen, Jan (2018). *Track changes*. URL: https://wiki.documentfoundation.org/Track_changes.

Jaro, Matthew A (1989). "Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida". In: *Journal of the American Statistical Association* 84.406, pp. 414–420.

Jurafsky, Dan and James H Martin (2014). *Speech and language processing*. Pearson.

Kamp, Hans, Josef Van Genabith, and Uwe Reyle (2011). "Discourse representation theory". In: *Handbook of philosophical logic*. Springer, pp. 125–394.

Kintsch, Walter and Teun A Van Dijk (1978). "Toward a model of text comprehension and production." In: *Psychological review* 85.5, p. 363.

Kouylekov, Milen and Bernardo Magnini (2005). "Recognizing textual entailment with tree edit distance algorithms". In: *Proceedings of the First Challenge Workshop Recognising Textual Entailment*, pp. 17–20.

Krippendorff, Klaus (2011). "Computing Krippendorff's alpha-reliability". In:

Lee, Ming Che, Jia Wei Chang, and Tung Cheng Hsieh (2014). "A grammar-based semantic similarity algorithm for natural language sentences". In: *The Scientific World Journal* 2014.

Levenshtein, Vladimir I (1966). "Binary codes capable of correcting deletions, insertions and reversals". In: *Soviet physics doklady*. Vol. 10, p. 707.

Lever, Jake, Martin Krzywinski, and Naomi Altman (2016). *Points of significance: classification evaluation*.

Li, Yuhua et al. (2006). "Sentence similarity based on semantic nets and corpus statistics". In: *IEEE transactions on knowledge and data engineering* 18.8, pp. 1138–1150.

Limited, StatsDirect (2016). *Agreement of Categorical Measurements*. URL: http://www.statsdirect.com/help/agreement/kappa.htm.

Lin, Dekang (1998). "An information-theoretic definition of similarity." In: *ICML*. Vol. 98. Citeseer, pp. 296–304.

Liu, Chang, Daniel Dahlmeier, and Hwee Tou Ng (2010). "TESLA: Translation evaluation of sentences with linear-programming-based analysis". In: *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*. Association for Computational Linguistics, pp. 354–359.

Liu, Fei et al. (2014). "A step towards usable privacy policy: Automatic alignment of privacy statements". In:

Lu, Jiaheng et al. (2013). "String similarity measures and joins with synonyms". In: *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. ACM, pp. 373–384.

MacCartney, Bill et al. (2006). "Learning to recognize features of valid textual entailments". In: *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*. Association for Computational Linguistics, pp. 41–48.

MacKenzie, David, Paul Eggert, and Richard Stallman (2003). *Comparing and Merging Files with GNU diff and patch*. Network Theory Ltd.

Madnani, Nitin, Joel Tetreault, and Martin Chodorow (2012). "Re-examining machine translation metrics for paraphrase identification". In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, pp. 182–190.

Magnini, Bernardo et al. (2014). "The Excitement Open Platform for Textual Inferences." In: *ACL (System Demonstrations)*, pp. 43–48.

Maloney, Erin (2003). "Values of the Pearson Correlation". In:

Μαλακασιώτης, Πρόδρομος (2011). "Paraphrasing and textual entailment recognition and generation". PhD thesis. Οικονομικό Πανεπιστήμιο Αθηνών. Τμήμα Πληροφορικής.

Marzal, Andres and Enrique Vidal (1993). "Computation of normalized edit distance and applications". In: *IEEE transactions on pattern analysis and machine intelligence* 15.9, pp. 926–932.

McCowan, Iain et al. (2004). "On the use of information retrieval measures for speech recognition evaluation". In: *Idiap-RR Idiap-RR-73-2004, IDIAP, Martigny, Switzerland, 0*.

McCready, Eric, Katsuhiko Yabushita, and Kei Yoshimoto (2014). *Formal Approaches to Semantics and Pragmatics: Japanese and Beyond*. Vol. 95. Springer.

McKeown, Kathleen R (1979). "Paraphrasing using given and new information in a question-answer system". In: *Proceedings of the 17th annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 67–72.

McWilliams, Jenna et al. (2013). "Using collaborative writing tools for literary analysis: Twitter, fan fiction and the crucible in the secondary English classroom". In: *Journal of Media Literacy Education* 2.3, p. 5.

Metzler, Donald, Susan Dumais, and Christopher Meek (2007). "Similarity measures for short segments of text". In: *European Conference on Information Retrieval*. Springer, pp. 16–27.

Mihalcea, Rada, Courtney Corley, and Carlo Strapparava (2006). "Corpus-based and knowledge-based measures of text semantic similarity". In: *AAAI*. Vol. 6, pp. 775–780.

Miller, George A (2009). "Wordnet-about us". In: *WordNet. Princeton University*.

Mozafari, Niloofar, Sattar Hashemi, and Ali Hamzeh (2011). "A precise statistical approach for concept change detection in unlabeled data streams". In: *Computers & Mathematics with Applications* 62.4, pp. 1655–1669.

Mulligan, Christopher and R Garofalo (2011). "A collaborative writing approach: Methodology and student assessment". In: *The Language Teacher* 35.3, pp. 5–10.

Myers, Eugene W (1986). "AnO (ND) difference algorithm and its variations". In: *Algorithmica* 1.1-4, pp. 251–266.

Navarro, Gonzalo (2001). "A guided tour to approximate string matching". In: *ACM computing surveys (CSUR)* 33.1, pp. 31–88.

Nelken, Rani and Stuart M Shieber (2006). "Towards Robust Context-Sensitive Sentence Alignment for Monolingual Corpora." In: *EACL*.

Neuwirth, Christine M et al. (1992). "Flexible diff-ing in a collaborative writing system". In: *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*. ACM, pp. 147–154.

Noël, Sylvie and Jean-Marc Robert (2004). "Empirical study on collaborative writing: What do co-authors do, use, and like?" In: *Computer Supported Cooperative Work (CSCW)* 13.1, pp. 63–89.

Nordquist, Richard (2016). *Imperative Sentence (Grammar)*. URL: http://grammar.about.com/od/il/g/impersent09.htm.

Pakray, Partha (2011). "Answer validation through textual entailment". In: *International Conference on Application of Natural Language to Information Systems*. Springer, pp. 324–329.

Parker, Kevin R and Joseph T Chao (2007). "Wiki as a teaching tool". In: *Interdisciplinary journal of knowledge and learning objects* 3.1, pp. 57–72.

Patil, MS, MS Bewoor, and SH Patil (2014). *Survey on Extractive Text Summarization Approaches*.

Pazienza, Maria Teresa, Marco Pennacchiotti, and Fabio Massimo Zanzotto (2005). "A linguistic inspection of textual entailment". In: *Congress of the Italian Association for Artificial Intelligence*. Springer, pp. 315–326.

Pham, Nghia et al. (2013). "Sentence paraphrase detection: When determiners and word order make the difference". In: *Proceedings of the Towards a Formal Distributional Semantics Workshop at IWCS 2013*, pp. 21–29.

Piolat, Annie (1991). "Effects of word processing on text revision". In: *Language and Education* 5.4, pp. 255–272.

Pustejovsky, James and Amber Stubbs (2012). *Natural language annotation for machine learning*. " O'Reilly Media, Inc."

Regneri, Michaela, Rui Wang, and Manfred Pinkal (2014). "Aligning Predicate-Argument Structures for Paraphrase Fragment Extraction." In: *LREC*, pp. 4300–4307.

Rieck, Konrad and Christian Wressnegger (2016). "Harry: a tool for measuring string similarity". In: *Journal of Machine Learning Research* 17.9, pp. 1–5.

Romano, Lorenza et al. (2006). "Investigating a Generic Paraphrase-Based Approach for Relation Extraction." In: *EACL*.

Sammons, Mark, V Vydiswaran, and Dan Roth (2011). "Recognizing textual entailment". In: *Multilingual Natural Language Applications: From Theory to Practice. Prentice Hall, Jun*.

Sanchez-Perez, Miguel A, Grigori Sidorov, and Alexander F Gelbukh (2014). "A Winning Approach to Text Alignment for Text Reuse Detection at PAN 2014." In: *CLEF (Working Notes)*, pp. 1004–1011.

Scheliga, Kaja (2015). "Collaborative writing in the context of science 2.0". In: *Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business*. ACM, p. 30.

SchÃűch, Christof (2014). *The right tool for the job: Five collaborative writing tools for academics*. URL: http://blogs.lse.ac.uk/impactofsocialsciences/2014/04/04/five-collaborative-writing-tools-for-academics/.

Sha, Lei et al. (2015). "Recognizing Textual Entailment Using Probabilistic Inference". In: 1620–âĂŞ1625.

Sharples, Mike et al. (1993). "Research issues in the study of computer supported collaborative writing". In: *Computer supported collaborative writing*. Springer, pp. 9–28.

Shinyama, Yusuke and Satoshi Sekine (2003). "Paraphrase acquisition for information extraction". In: *Proceedings of the second international workshop on Paraphrasing-Volume 16*. Association for Computational Linguistics, pp. 65–71.

Shnarch, Eyal (2008). *Lexical entailment and its extraction from Wikipedia*. Bar Ilan University, Department of Mathematics and Computer Science.

Sokolova, Marina and Guy Lapalme (2009). "A systematic analysis of performance measures for classification tasks". In: *Information Processing & Management* 45.4, pp. 427–437.

Southavilay, Vilaythong et al. (2013). "Analysis of collaborative writing processes using revision maps and probabilistic topic models". In: *Proceedings of the Third International Conference on Learning Analytics and Knowledge*. ACM, pp. 38–47.

Standards director, Society for Editors and Proofreaders Ltd (2016). *FAQs: What is copy-editing?* URL: http://www.sfep.org.uk/about/faqs/what-is-copy-editing/.

Stern, Asher and Ido Dagan (2014). "The biutee research platform for transformation-based textual entailment recognition". In: *LiLT (Linguistic Issues in Language Technology)* 9.

Stern, Asher et al. (2012). "Efficient search for transformation-based inference". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, pp. 283–291.

Storch, Neomy (2005). "Collaborative writing: Product, process, and studentsâĂŹ reflections". In: *Journal of second language writing* 14.3, pp. 153–173.

Straus, Jane, Lester Kaufman, and Tom Stern (2014). *The blue book of grammar and punctuation: An easy-to-use guide with clear rules, real-world examples, and reproducible quizzes*. John Wiley & Sons.

Tatar, Doina et al. (2009). "Textual entailment as a directional relation". In: *Journal of Research and Practice in Information Technology* 41.1, p. 53.

Toledo, Assaf et al. (2013). "Semantic annotation of textual entailment". In: *Proceedings of the 10th International Conference on Computational Semantics (IWCS 2013)–Long Papers*. Citeseer, pp. 240–251.

TREnTIn, Guglielmo (2009). "Using a wiki to evaluate individual contribution to a collaborative learning project". In: *Journal of computer assisted learning* 25.1, pp. 43–55.

Van Asch, Vincent (2013). "Macro-and micro-averaged evaluation measures [[basic draft]]". In: *Belgium: CLiPS*.

Van Dijk, Teun A (1977). "Semantic macro-structures and knowledge frames in discourse comprehension". In: *Cognitive processes in comprehension*, pp. 3–32.

Van Dijk, Teun Adrianus (1980). *Macrostructures: An interdisciplinary study of global structures in discourse, interaction, and cognition*. Lawrence Erlbaum Associates.

Vo, Ngoc Phuoc An, Simone Magnolini, and Octavian Popescu (2015). "Paraphrase identification and semantic similarity in twitter with simple features". In: *The 3rd International Workshop on Natural Language Processing for Social Media*, p. 10.

Wallace, David L and John R Hayes (1991). "Redefining revision for freshmen". In: *Research in the Teaching of English*, pp. 54–66.

Wallis, Peter (1993). "Information retrieval based on paraphrase". In: *Proceedings of PACLING Conference*. Citeseer.

Wallis, Sean (2007). "Annotation, Retrieval and Experimentation". In: *Annotating Variation and Change. Helsinki: Varieng,[University of Helsinki]*.

Wang, Rui and Chris Callison-Burch (2011). "Paraphrase fragment extraction from monolingual comparable corpora". In: *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*. Association for Computational Linguistics, pp. 52–60.

Wang, Rui and Günter Neumann (2007). "Recognizing textual entailment using a subsequence kernel method". In: *AAAI*. Vol. 7, pp. 937–945.

Wang, Yuan, David J DeWitt, and J-Y Cai (2003). "X-Diff: An effective change detection algorithm for XML documents". In: *Data Engineering, 2003. Proceedings. 19th International Conference on*. IEEE, pp. 519–530.

Watanabe, Yotaro et al. (2013). "Overview of the Recognizing Inference in Text (RITE-2) at NTCIR-10." In: *NTCIR*.

Weiss, Stéphane, Pascal Urso, and Pascal Molli (2007). "Wooki: a p2p wiki-based collaborative writing tool". In: *International Conference on Web Information Systems Engineering*. Springer, pp. 503–512.

Wijaya, Derry Tanti and Reyyan Yeniterzi (2011). "Understanding semantic change of words over centuries". In: *Proceedings of the 2011 international workshop on DETecting and Exploiting Cultural diversiTy on the social web*. ACM, pp. 35–40.

Winkler, William E (1990). "String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage." In:

Wołk, Krzysztof and Krzysztof Marasek (2014). "A Sentence Meaning Based Alignment Method for Parallel Text Corpora Preparation". In: *New Perspectives in Information Systems and Technologies, Volume 1*. Springer, pp. 229–237.

Xu, Yong, Aurélien Max, and François Yvon (2015). "Sentence alignment for literary texts". In: *LiLT (Linguistic Issues in Language Technology)* 12.

Yang, Diyi et al. (2017). "Identifying semantic edit intentions from revisions in wikipedia". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2000–2010.

Yarrow, Fiona and Keith J Topping (2001). "Collaborative writing: The effects of metacognitive prompting and structured peer interaction". In: *British journal of educational psychology* 71.2, pp. 261–282.

Zanzotto, Fabio Massimo and Marco Pennacchiotti (2010). "Expanding textual entailment corpora from Wikipedia using co-training". In: *Proceedings of the COLING-Workshop on The PeopleâĂŹs Web Meets NLP: Collaboratively Constructed Semantic Resources*. Vol. 128.

Zhang, Fan and Diane Litman (2014). "Sentence-Level Rewriting Detection." In: *Grantee Submission*.

— (2015). "Annotation and classification of argumentative writing revisions". In: *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 133–143.

Zhang, Maoyuan et al. (2014). "Sentence Level Paraphrase Recognition Based on Different Characteristics Combination". In: *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*. Springer, pp. 279–289.

Zhang, Weinan et al. (2015). "Exploring Key Concept Paraphrasing Based on Pivot Language Translation for Question Retrieval." In: *AAAI*, pp. 410–416.

Zhao, Shiqi and Haifeng Wang (2010). "Paraphrases and applications". In: *Proceedings of the 23rd International Conference on Computational Linguistics: Tutorial notes: Paraphrases and applications*. Association for Computational Linguistics, pp. 1–87.

Zhou, Wenyi, Elizabeth Simpson, and Denise Pinette Domizi (2012). "Google Docs in an Out-of-Class Collaborative Writing Activity." In: *International Journal of Teaching and Learning in Higher Education* 24.3, pp. 359–375.

Appendix A

Author Feedback Form

Survey: Significant Changes between Versioned Text Documents

Introduction

Thank you very much for taking the time and effort for this survey.

This survey is conducted for the research purposes of understanding how authors judge the significance of changes between versioned text documents they have written and revised.

Your feedback is highly appreciated. The information gathered in this survey will help us to derive:

- An operational definition of significant changes between versioned text documents
- A representation of changes between versioned text documents

which will further assist us in developing a method to automatically identify significant changes between versioned text documents.

This survey will take ***approximately one hour*** to complete.

There are three sections of this survey:

A. Change Identification
B. Defining of Local and Global Changes
C. The Effect of Grouping Changes

Further instructions will be provided at the start of each section.

If you have any queries, kindly please contact Ping Ping Tan (email: *pingt@student.unimelb.edu.au*)

Section A: Change Identification

Instruction:

We will provide text changes extracted from the versioned text documents you have authored and revised. Each of the questions require you to identify the changes, to indicate whether the change results in any meaning change and for you to rate the impact of the changes. For questions that require written answers, an example of an answer is provided as a guide.

1. For the example below:

Original	*Revision*
Label pathology on X-ray	Label pathology on Annotated X-ray

a) List the changes in the revised version. (*Example: added the word Annotated*)
Click here to enter text.

b) Do you consider that the change above results in any **meaning change**?

☐	YES	☐	NO

c) Based on the change(s) you list in a)

(i) How do you rate the impact of the change and (*Example: no change or minor change or major change*)
Click here to enter text.

(ii) Provide a justification for the rating.
Click here to enter text.

2. For the example below, using the notation below:

Notation
[D:] – Delete [A:] – Add [S: original word -> new word] – Substitute

Original	*Revision*
UC2.4 Label Pathology 1. Select an image suitable for labelling with pathology information 2. Label pathology on X-ray 3. Provide a text identifier and save labelled image with Current Patient Information	**Label Pathology** 1. Select an [S: image -> Annotated X-ray] suitable for labelling with pathology information 2. Label pathology on [A: Annotated] X-ray includes suggestions] 3. Provide a text identifier and save labelled [S: image -> Annotated X-ray] with Current Patient [S: Information -> Record]

a) For each of the changes listed in the table below, do you consider the change results in any **meaning change**? How do you rate the impact of the change? Justify your answer.

Change	Meaning Change?	Impact of Change and Justification
(Example: Number 1 - Substitute the word image to Annotated X-ray)	*Example:* ☒ YES ☐ NO	*(Example: Major change, image and Annotated X-ray are different)*
1 - Substitute the word *image to Annotated X-ray*	☐ YES ☐ NO	Click here to enter text.
2 - Added the word *Annotated*	☐ YES ☐ NO	Click here to enter text.
2 - Added the statement *Section 3.1 Predefined Labels includes suggestion*	☐ YES ☐ NO	Click here to enter text.
3 - Substitute the word *image* to *Annotated X-ray*	☐ YES ☐ NO	Click here to enter text.
3 - Substitute the word *Information* to *Record*	☐ YES ☐ NO	Click here to enter text.

b) Do you prefer the changes to be highlighted like in this example compared to the example in Question 2? Justify your answer.

☐	YES	☐	NO	*(Example of Justification: it improves readability)* Click here to enter text.

c) Out of the 5 changes listed for this example, rank the significance of each change from the most significant to the least significant (*two or more of the changes can have the same*

significant, 1 with the most significant and 5 the least significant). Provide a justification for your ranking.

Change	Significant Rank	Justification
(Example:1 - Substitute the word image to Annotated X-ray)	*(Example: 1)*	*(Example: Image and Annotated X-ray are different)*
1 - Substitute the word *image* to *Annotated X-ray*	Click here to enter text.	Click here to enter text.
2 - Added the word *Annotated*	Click here to enter text.	Click here to enter text.
2 - Added the statement *Section 3.1 Predefined Labels includes suggestion*	Click here to enter text.	Click here to enter text.
3 - Substitute the word *image* to *Annotated X-ray*	Click here to enter text.	Click here to enter text.
3 - Substitute the word *Information* to *Record*	Click here to enter text.	Click here to enter text.

d) Based on the **minor changes** you had rated in Question a),

(i) If the minor changes on *improvement of style or readability* (i.e. no meaning change) are considered together, will these changes become a significant change (i.e. major impact change)? *If YES, state the number of minor changes that would result in a significant change and provide a justification for that number?*

☐	YES	☐	NO	(*Example: three because the three changes grouped together affect the structure of the use case*) Click here to enter text.

(ii) If the minor changes on *meaning change* are considered together, will these changes become a significant change (i.e. major impact change)? *If YES, state the number of minor changes and provide a justification for that number?*

☐	YES	☐	NO	(*Example: three because the three changes grouped together affect the meaning of the use case*) Click here to enter text.

(iii) Taking into both types of minor changes, how many minor changes (*i.e. number of changes, zero is an acceptable number*) will add up to be a significant change (i.e. major impact change)? State your justification for the number provided *(Example: two minor changes on improvement of style or readability and one minor meaning change because these changes changed the overall meaning)*.

Click here to enter text.

e) Can significant change be directly equated to major change? Provide a justification of your answer.

☐	YES	☐	NO	(*Example: YES because minor and major changes are an indicator of significance*) Click here to enter text.

Section B: Defining Local and Global Changes

Instruction:

Using the notation as below, first, you should assess the impact of change involved in each of cases.

Notation
[D:] – Delete
[A:] – Add
[S: original word -> new word] – Substitute

a) Rate the impact of each change (*Check the checkbox, only one option is allowed*).
b) Justify how significant the change is.
c) State how you believe the change should be represented.

An ***example*** is provided below:

Version 0.9
Display Completed Schedule The system displays the schedule containing the selected course offerings for the Student and the confirmation number for the schedule.

Change	Version 1.0
1	Display [D: Completed] Schedule
2	The system displays the schedule containing the selected course offerings for the Student and the [S: confirmation number for the schedule -> *reminder to attend the first class for each course in order to complete the registration for the course*.]
3	[A: Send Completed Schedule, **include** (Attend First Class) The system receives the Student First Class Attendance and the system sends to the Student's email the confirmation for the course registration.]

Change	Rate the Impact of Change		Significance Justification	Change Representation
1	None	☐	*(Example: Minor significance, Step 6, no longer need to produce complete schedule, as it is the use case name, not much change but the changes in the content is important.)*	*(Example: Delete Completed from Completed Schedule.)*
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☒		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
2	None	☐	*(Example: Minor significance, A*	*(Example: Show a screen to differentiate*
	Minor change (improvement	☒		

	to style or readability)		*component had been changed to another component.)*	*between the original statement and another to show where had been deleted.)*
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
3	None	☐	*(Example: Significant, A 'include (Attend First Class)' use case has been added.)*	*(Example: Send a warning sign.)*
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☒		

Case 1: Basic Flow

Version 0.9
Start-up Invoke OWS Software licence checking, if any Surgeon authentication, e.g. user id and password, may be performed for safety and data security reasons

Change	Version 1.0
	UC2.1 Start-up
	UC2.1.1 Invoke OWS
	UC2.1.2 Software licence checking, if any
1	UC2.1.3 [D: Surgeon] Authentication, e.g. user id and password,
2	[S: maybe -> is] performed for safety and data security reasons
3	[A: There are privacy issues related to Patient Details, including X-rays, and there is a possibility that the system could be deployed in a multi user environment.]

Change	Rate the Impact of Change		Significance Justification	Change Representation
1	None	☐	Click here to enter text.	Click here to enter text.

	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
2	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
3	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		

Case 2: Basic Flow

Version 0.9
Load X-Rays Indicate location of x-rays. Check X-Rays are for Current Patient Store X-Ray with Current Patient information

Change	Version 1.0

1	**Load [A: OWS]**
2	**X-Ray[D: s]**
3	Indicate location of [A: OWS]
4	X-ray[D: s].
5	Check [A: OWS] X-Ray are for Current Patient
6	Store [A: OWS] X-Ray
7	[A: as Annotated X-ray] with
8	Current Patient [S: Information -> Record].
9	[A: As OWS X-rays are added to the Current Patient Record, they become Annotated X-rays (even if there are no annotations added yet). The Annotated X-rays will be uniquely identifiable for each Patient.
	This use case will need to be repeated for each OWS X-ray loaded for the Current Patient]

Change	Rate the Impact of Change		Significance Justification	Change Representation
1	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
2	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
3	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
4	None	☐	Click here to enter text.	Click here to enter text.

	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
5	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
6	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
7	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
8	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		

9	None	☐	Click here to enter text.	Click here to enter text.

	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		

Case 3: Basic Flow

Version 0.9
Label Pathology Select an image suitable for labelling with pathology information Label pathology on X-ray Provide a text identifier and save labelled image with Current Patient Information

Change	Version 1.0
	Label Pathology
1	Select an [S: image -> Annotated X-ray] suitable for labelling with pathology information
	Label pathology on [A: Annotated] X-ray
2	[A: Section 3.1 Predefined Labels includes suggestions]
3	Provide a text identifier and save labelled [S: image -> Annotated X-ray] with
4	Current Patient [S: Information -> Record]
5	

Change	Rate the Impact of Change		Significance Justification	Change Representation
1	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
2	None	☐	Click here to enter text.	Click here to enter text.

	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
3	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
4	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
5	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		

Case 4: Basic Flow

Version 0.9
Develop Composite ***Initialise Composite*** Select an "AP pelvis" X-ray Set Side (left or right) for Hip-Replacement Set IRType for Hip-Replacement Annotate X-ray with line to indicate scale: this could be done manually, or with automated assistance based on image processing Set the scale by indicating that this is the scale line and by providing the length of this scale line

Change	Version 1.0
1 2	**Develop Composite** ***Initialise Composite*** Select an "AP pelvis" [A: Annotated] X-ray Set Side (left or right) for Hip-Replacement [D: UC2.5.15.5 Set IRType for Hip-Replacement] Annotate X-ray with line to indicate scale: this could be done manually, or with automated assistance based on image processing Set the scale by indicating that this is the scale line and by providing the length of this scale line

Change	Rate the Impact of Change		Significance Justification	Change Representation
1	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
2	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		

Case 5: Basic Flow

Version 0.9
Identify the maximum medial points of the lesser trochanters.

Change	Version 1.0
1	Identify the maximum medial points [A: (or tops)] of the lesser trochanters.

Change	Rate the Impact of Change		Significance Justification	Change Representation
1	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		

Case 6: Basic Flow

Version 0.9
Identify CORS of Non-destroyed Hip and Target AS Diameter
Annotate X-ray with radial line to indicate CORS and socket radius: this could be done manually (selecting centre and increasing radius of circle, or selecting three points on circumference) or with automated assistance based on image processing.

Change	Version 1.0
1	***Identify CORS of [S: non-destroyed -> normal (Contra-lateral)] Hip and Target AS Diameter***
2	[A: This is not necessarily the same as centre of rotation of the head, but is the centre of rotation of the acetabulum or more likely the desired centre of rotation of the replaced hip]
	Annotate X-ray with radial line to indicate CORS and socket radius: this could be done manually (selecting centre and increasing radius of circle, or selecting three points on circumference) or with automated assistance based on image processing.
3	[A: The diameter of the circle indicated by the radial line is used when making an initial selection of the AS]

Change	Rate the Impact of Change		Significance Justification	Change Representation
1	None	☐	Click here to enter text.	Click here to enter text.

	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
2	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
3	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		

Case 7: Basic Flow

Version 0.9
Calculate Offset of non-destroyed Hip Identify the longitudinal axis of the femur: i. identify at least two pairs of points on the outer edge of the femur ii. bisect the distance between each pair of points iii. draw a line through those bisection points.

Change	Version 1.0
1 2 3 4	***Calculate Offset of [S: non-destroyed -> normal (Contra-lateral)] Hip*** Identify the longitudinal axis of the femur: i. identify [D: at least] two pairs of points on the outer edge [A: (cortex)] of the femur: [A: (one pair just inferior to the lesser trochanter and one pair at the lowest visible region on the X-ray)] ii. bisect the distance between each pair of points iii. draw a line through those bisection points.

Change	Rate the Impact of Change		Significance Justification	Change Representation
1	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
2	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
3	None	☐	Click here to enter text.	Click here to enter text.

	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
4	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		

Case 8: Basic Flow

Version 0.9
Identify Replacement Parameters of Destroyed HipReflect CORS radial line of non-destroyed hip onto destroyed hip Draw cut-line on femur: i. Draw line from top of lesser trochanter to the bottom of the head of the femur.

Change	Version 1.0
1	***Identify Replacement Parameters of [S: Destroyed -> Diseased] Hip***
2	Reflect CORS radial line of [S: non-destroyed -> normal] hip onto
3	[S: destroyed -> diseased (ipsi-lateral)] hip
4	Draw [S: cut- -> initial femoral resection]line[I:)] on femur: i. Draw line from top of lesser trochanter to the bottom of the head of the femur.

Change	Rate the Impact of Change	Significance Justification	Change Representation

1	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
2	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
3	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
4	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		

Case 9: Basic Flow

Version 0.9
Select Shell Select Initial Acetabular Shell: Default is IRType, but can be overridden. Size biased towards preset AS diameter. Change the template until Surgeon is satisfied (selecting different diameters of Shell). Select material for Insert

Change	Version 1.0
1	***Select Shell [A: The use enters "Select Shell" mode ie. The AS template is added as an annotation to the X-ray. The user is able only to perform functions related to the selection of the AS. Confirmation of completion of AS selection is required.]***
2	[A: Select Intended Replacement Type for Acetabular Shell (IRType(AS)]
3	Select Initial Acetabular Shell: Default is IRType[A: (AS)], but can be overridden. Size biased towards preset AS diameter. Change the template until Surgeon is satisfied (selecting different diameters of Shell).
4	Select material [S: for -> , internal diameter, and other attributes eg. low profile, extended rim, of] Insert

Change	Rate the Impact of Change		Significance Justification	Change Representation
1	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
2	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
3	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to	☐		

	style or readability)			
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
4	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		

Case 10: Basic Flow

Version 0.9
Select Femoral Stem ***Select Initial Femoral Stem: Default is IRType, but can be overridden. Stem offset biased towards preset Offset. Stem width initially narrowest.***

Change	Version 1.0
1	***Select Femoral Stem [A:*** Select Intended Replacement Type for Femoral System (IRType(FS)]Select Initial Femoral Stem: Default is IRType[A: (FS)], but can be overridden. Stem offset biased towards preset Offset. Stem width initially narrowest.
2	
3	

Change	Rate the Impact of Change		Significance Justification	Change Representation
1	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
2	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement	☐		

	to style or readability)			
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
3	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		

Case 11: Basic Flow

Version 0.9
Select Femoral Head: size and material (which must be compatible with Insert size and material)

Change	Version 1.0
1	Select Femoral Head: neck length, size, and material ([D: which] must be compatible with Insert size and material)

Change	Rate the Impact of Change		Significance Justification	Change Representation
1	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		

Case 12: Basic Flow

Version 0.9
Adjust femur cut-line

Change	Version 1.0
1	Adjust [S: femur cut- -> femoral neck resection] line

Change	Rate the Impact of Change		Significance Justification	Change Representation
1	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		

Case 13: Basic Flow

Version 0.9
The surgeon can supply confidence factors for individual components in the plan

Change	Version 1.0
1	The surgeon can supply confidence factors for individual components in the plan [A: Confidence factors may be used to determine the sizes of components delivered above and below the planned size]

Change	Rate the Impact of Change		Significance Justification	Change Representation
1	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		

Case 14: Basic Flow

Version 0.9
On completion of the above, the surgeon will request the system to generate the orders. These orders will be saved in a format suitable for printing or email. The system creates a set of orders, one for each supplier corresponding to a component or accessory specified in the Operation Plan. All components sourced from a particular supplier will appear in that supplier's order, as will other relevant details (such as confidence levels) from the Operation Plan

Change	Version 1.0
1	On completion of the above, the surgeon will request the system to generate the orders. These orders will be saved in a format suitable for printing or email. The system creates a set of orders, one for each supplier corresponding to a component or accessory specified in the Operation Plan. All components sourced from a particular supplier will appear in that supplier's order, as will other relevant details (such as confidence levels) from the Operation Plan. [A: Copies of the orders are also sent to the hospital]

Change	Rate the Impact of Change		Significance Justification	Change Representation
1	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		

Case 15: Alternative Flows

Version 0.9
Performing pre-operative planning using X-ray of hip

Change	Version 1.0
1	Performing pre-operative planning using [A: Ipsi-lateral]
2	X-ray of hip [A: only]

Change	Rate the Impact of Change		Significance Justification	Change Representation
1	None	☐	Click here to enter text.	Click here to enter text.

	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		
2	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		

Case 16: Alternative Flows

Version 0.9
2.1.5.2 Determine X-ray LLD, steps 1-4. The working LLD can be recorded based only on the observed LLD

Change	Version 1.0
1	2.1.5.2 Determine X-ray LLD, steps 1-4. The working LLD can be recorded based only on the [A: clinically] observed LLD

Change	Rate the Impact of Change		Significance Justification	Change Representation
1	None	☐	Click here to enter text.	Click here to enter text.
	Minor change (improvement to style or readability)	☐		
	Minor change (meaning change)	☐		
	Major change (improvement to style or readability)	☐		
	Major change (meaning change)	☐		

Section C: The Effect of Grouping Changes

Instruction:

This section is based on your ratings in Section B. Each of the questions requires you to identify the changes when the changes are grouped together and select the best answer representing the changes. For questions that require written answers, an example of an answer is provided as a guide.

1. Based on the rating you had chosen for the **minor changes** in **Section B**,
   a) Do you group local minor changes on *improvement of style or readability* (i.e. no meaning change) will result to become a significant change to the overall revised specification?

☐	YES	☐	NO

   b) Do you group minor changes on *meaning change* will result to become a significant change to the overall revised specification?

☐	YES	☐	NO

   c) Taking into both types of minor changes, how many minor changes (*i.e. number of changes, zero is an acceptable number*) will add up to be a significant change (i.e. major impact change) in the overall revised specification? State your justification for the number provided *(Example: two minor changes on improvement of style or readability and one minor meaning change because these changes changed the overall meaning)*.
   Click here to enter text.

2. Based on the **major changes** you had chosen in **Section B**,
   a) Can we directly equate the major changes on improvement of style or readability (i.e. no meaning change) as significant changes in the overall revised specification?

☐	YES	☐	NO

   b) Can we directly equate the major changes on meaning change as significant changes in the overall revised specification?

☐	YES	☐	NO

3. Based on the ***minor and major changes*** you had chosen in Section I,
   a) When minor and major changes of *style or readability* (i.e. no meaning change) are grouped, will it result in a **significant change** to the overall revised specification?

☐	YES	☐	NO

   b) When minor and major changes of *meaning* are grouped, will it result in a **significant change** to the overall revised specification?

☐	YES	☐	NO

   c) When minor and major changes of *both types* (i.e. with or without meaning changes) are grouped, will it result in a **significant change** to the overall revised specification?

☐	YES	☐	NO

4. At what point would you consider that the changes are significant enough to create the next version of the specification? Provide a justification for your answer. (*Example: based on the number of changes agreed as the number is an approved value by the team*)
   Click here to enter text.

Thank you very much

Appendix B

Non-author Feedback Form

Significant Changes between Revised Text Documents

Welcome to Significant Revision Changes Questionnaire. Kindly please review the Plain Language Statement below and if you agree to it, press Continue at the bottom of the page to proceed.

* Required

PLAIN LANGUAGE STATEMENT

THE UNIVERSITY OF MELBOURNE

"Significant Changes between Versioned Text Documents"

You are invited to participate in the above research project, which is being conducted by Associate Professor Dr. Karin Verspoor (principal supervisor), Dr. Timothy Miller (co-supervisor) and Ms. Tan, Ping Ping (PhD student) of the Department of Computing and Information Systems at The University of Melbourne. This project will form part of Ms. Tan's PhD thesis, and has been approved by the Human Research Ethics Committee.

The aim of this study is to understand how people judge the significance of changes between versions of a text document. Text changes extracted from versioned text documents will be provided to you. Should you agree to participate, you would be asked to look at the text changes and complete a 15 minute questionnaire, at a time convenient for you. This questionnaire will ask you to rate the significance of individual changes in terms of the following scale: Major Change, Minor Change or No Change. We estimate that the time commitment required of you would not exceed 30 minutes.

We intend to protect your anonymity and the confidentiality of your responses to the fullest possible extent, within the limits of the law. You are only required to provide a yes or no answer to ensure that you are of the age 18 and above. Even though you may have received this questionnaire via email or social media link, we do not keep any personal details (other than age) and the questionnaire will be submitted as an anonymous survey.

Once the thesis arising from this research has been completed, a brief summary of the findings will be made available by researchers upon application. The results of this study will be reported as group data only. Your individual information will not be identifiable in the report. To further protect your confidentiality and anonymity, we will store your name and contact details in a separate, locked cabinet from the data you supply. All computer files will be accessible to the researchers only, and will be password protected. You should note that these measures are only able to guarantee confidentiality within the limits of the law. It is also possible that the results will be presented at academic conferences. The data will be kept securely in the Department of Computing and Information Systems for five years from the date of publication, before being destroyed.

Please be advised that your participation in this study is completely voluntary. Should you wish to withdraw at any stage, or to withdraw any unprocessed data you have supplied, you are free to do so without prejudice. The researchers are not involved in the ethics application process. Your decision to participate or not, or to withdraw, will be completely independent of your dealings with the ethics committee, and we would like to assure you that it will have no effect on any applications for approval that you may submit.

If you would like to participate, please indicate that you have read and understood this information. By clicking on the "Continue" button, you automatically give your consent to participate in the questionnaire.

Should you require any further information, or have any concerns, please do not hesitate to contact the responsible researcher; AP Dr Verspoor: 8344 4902 (email: karin.verspoor@unimelb.edu.au). Should you have any concerns about the conduct of the project, you are welcome to contact the Executive Officer, Human Research Ethics, The University of Melbourne, on ph: 8344 2073, or fax: 9347 6739.

Department of Computing and Information Systems
The University of Melbourne, Victoria 3010 Australia
T: +61 3 8344 1501 **F:** +61 3 9349 4596
W: http://www.cis.unimelb.edu.au/

Human Ethics Application 1544698, 30 July, 2015

unimelb.edu.au

Thank you for participating in our questionnaire and we appreciate your time and effort in answering this questionnaire. It should take less than 20 minutes to answer all the items. Your feedback is important for the study to evaluate the computational method which we will develop later.

The current project aims to develop a computational method to identify significant changes between original and revised text documents. This survey is designed as a part of the project to collect user feedback on quantifying the impact of changes in revised text documents. The cases below are extracted from requirement specifications to develop an Orthopedic Workstation (OWS).

This survey consists of 12 cases with different changes (pairs of the texts before and after the revision) which were extracted from an actual revised specification. Read each pair and decide the impact of each of the change. Then mark the item according to the following scale:

- *Formal Change*:
  Revising the punctuation abbreviation, grammar, tense or spelling of a sentence, without changing the meaning of the sentence.
- *Meaning Preserving Change*:
  Rephrase or Reword to express the sentence in a different style that does not change the meaning of the sentence within the context

  (Example: "I paid a hundred dollars for the tickets to take my family to a movie." ->
  "I paid a hundred dollars to take my family to a movie.")
- *Minor Meaning Change*:
  Revision that alters the meaning of words within the sentence, BUT that does NOT alter the overall gist of the sentence in the greater context

  (Example: "I paid a hundred dollars for the tickets to take my family to a movie." ->
  "I paid a hundred dollars for the tickets, with popcorn and drinks, to bring my family to a movie.")
- *Major Meaning Change*:
  Revision to the sentence which alters the overall gist of the sentence in the greater context

  (Example: "I paid a hundred dollars for the tickets to bring my family to a movie." ->
  "We decided to watch movie at home.")

Answer all the items. Please work rapidly and do not spend too much time on any of the cases.

Case 1

Original	Revised
UC2.1 Start-up UC2.1.1 Invoke OWS. UC2.1.2 Software licence checking, if any. UC2.1.3 Surgeon authentication, e.g. user id and password, may be performed for safety and data security reasons.	**UC2.1 Start-up** UC2.1.1 Invoke OWS. UC2.1.2 Software license checking, if any. UC2.1.3 Authentication, e.g. user id and password, is performed for safety and data security reasons. There are privacy issues related to Patient Details, including X-Rays, and there is a possibility that the system could be deployed in a multi user environment.

1. **Rate all changes** *
   Mark only one oval per row.

	Formal Change	Meaning Preserving Change	Minor (Meaning change)	Major (Meaning Change)
UC2.1.3 Delete: 'Surgeon'	⬭	⬭	⬭	⬭
UC2.1.3 Substitute: 'may be' -> 'is'	⬭	⬭	⬭	⬭
Add: "There are privacy issues related to Patient Details, including X-Rays, and there is a possibility that the system could be deployed in a multi user environment."	⬭	⬭	⬭	⬭

Case 2

Original	Revised
UC2.3 Load X-Rays UC2.3.1 Indicate location of X-Rays. UC2.3.2 Check X-Rays are for Current Patient. UC2.3.3 Store X-Ray with Current Patient Information.	**UC2.3 Load OWS X-Ray** UC2.3.1 Indicate location of OWS X-Ray. UC2.3.2 Check OWS X-Ray is for Current Patient. UC2.3.3 Store OWS X-Ray as Annotated X-Ray with Current Patient Record. As OWS X-rays are added to the Current Patient Record, they become Annotated X-Rays (even if there are no annotations added yet). The Annotated X-Rays will be uniquely identifiable for each Patient. This use case will need to be repeated for each OWS X-Ray loaded for the Current Patient.

2. **Rate all changes** *
Mark only one oval per row.

	Formal Change	Meaning Preserving Change	Minor (Meaning change)	Major (Meaning Change)
UC2.3 Add: 'OWS'				
UC2.3 Delete: 's'				
UC2.3.1 Add: 'OWS'				
UC2.3.1 Delete: 's'				
UC2.3.3 Add: 'as Annotated X-ray'				
UC2.3.3 Substitute 'Information' -> 'Record'				
Add: "This use case will need to be repeated for each OWS X-Ray loaded for the Current Patient"				

Case 3

Original	Revised
UC2.4 Label Pathology	UC2.4 Label Pathology
UC2.4.1 Select an image suitable for labelling with pathology information.	UC2.4.1 Select an Annotated X-Ray suitable for labelling with pathology information.
UC2.4.2 Label pathology on X-Ray.	UC2.4.2 Label pathology on Annotated X-Ray. Section 3.1 Predefined Labels include suggestions.
UC2.4.3 Provide a text identifier and save labelled image with Current Patient Information.	UC2.4.3 Provide a text identifier and save labelled Annotated X-Ray with Current Patient Record.

3. **Rate all changes** *
Mark only one oval per row.

	Formal Change	Meaning Preserving Change	Minor (Meaning change)	Major (Meaning Change)
UC2.4.1 Substitute: 'image' -> 'Annotated X-Ray'				
UC2.4.2 Add: 'Annotated'				
Add: "Section 3.1 Predefined Labels include suggestions""				
Substitute: 'Information' -> 'Record'				

Case 4

Original	Revised
UC2.5 Develop Composite	**UC2.5 Develop Composite**
UC2.5.15 Initialise Composite	***UC2.5.15 Initialise Composite***
UC2.5.15.2 Select an "AP pelvis" X-Ray.	UC2.5.15.2 Select an "AP pelvis" Annotated X-Ray.
UC2.5.15.3 Set Side (left of right) for Hip-Replacement.	UC2.5.15.3 Set Side (left of right) for Hip-Replacement.
UC2.5.15.5 Set IRType for Hip-Replacement.	
UC2.5.15.4 Annotate X-Ray with line to indicate scale: this could be done manually, or with automated assistance based on image processing.	UC2.5.15.4 Annotate X-Ray with line to indicate scale: this could be done manually, or with automated assistance based on image processing.
UC2.5.15.1 Set the scale by indicating that this is the scale line and by providing the length of this scale line.	UC2.5.15.1 Set the scale by indicating that this is the scale line and by providing the length of this scale line.

4. **Rate all changes** *
Mark only one oval per row.

	Formal Change	Meaning Preserving Change	Minor (Meaning change)	Major (Meaning Change)
UC2.5.15.2 Add: 'Annotated'	◯	◯	◯	◯
UC2.5.15.5 Delete: "Set IRType for Hip-Replacement"	◯	◯	◯	◯

Case 5

Original	Revised
UC2.5.3.2 Identify the maximum medial points of the lesser trochanters.	UC2.5.3.2 Identify the maximum medial points (or tops) of the lesser trochanters.

5. **Rate all changes** *
Mark only one oval per row.

	Formal Change	Meaning Preserving Change	Minor (Meaning change)	Major (Meaning Change)
UC2.5.3.2 Add: '(or tops)'	◯	◯	◯	◯

Case 6

Original	Revised
UC2.5.4 Identify CORS of Non-destroyed Hip and Target AS Diameter	*UC2.5.4 Identify CORS of Normal (Contra-lateral) Hip and Target AS Diameter*
	This is not necessarily the same as centre of rotation of the head, but is the centre of rotation of the acetabulum or more likely the desired centre of rotation of the replaced hip.
UC2.5.4.1 Annotate X-Ray with radial line to indicate CORS and socket radius: this could be done manually (selecting centre and increasing radius of circle, or selecting three points on circumference) or with automated assistance based on image processing.	UC2.5.4.1 Annotate X-Ray with radial line to indicate CORS and socket radius: this could be done manually (selecting centre and increasing radius of circle, or selecting three points on circumference) or with automated assistance based on image processing.
	The diameter of the circle indicated by the radial line is used when making an initial selection of the AS.

6. **Rate all the changes** *

Mark only one oval per row.

	Formal Change	Meaning Preserving Change	Minor (Meaning change)	Major (Meaning Change)
UC2.5.4 Substitute: 'Non-destroyed' -> 'Normal (Contra-lateral)'	⬭	⬭	⬭	⬭
Add: "This is not necessarily the same as centre of rotation of the head, but is the centre of rotation of the acetabulum or more likely the desired centre of rotation of the replaced hip"	⬭	⬭	⬭	⬭
Add: "The diameter of the circle indicated by the radial line is used when making an initial selection of the AS"	⬭	⬭	⬭	⬭

Case 7

Original	Revised
UC2.5.5 Calculate Offset of Non-destroyed Hip	*UC2.5.5 Calculate Offset of Normal (Contra-lateral) Hip*
UC2.5.5.1 Identify the longitudinal axis of the femur:	UC2.5.5.1 Identify the longitudinal axis of the femur:
i. identify at least two pairs of points on the outer edge of the femur.	i. identify two pairs of points on the outer edge (cortex) of the femur (one pair just inferior to the lesser trochanter and one pair at the lowest visible region on the X-Ray).
ii. bisect the distance between each pair of points.	ii. bisect the distance between each pair of points.
iii. draw a line through those bisection points.	iii. draw a line through those bisection points.

7. **Rate all changes** *

Mark only one oval per row.

	Formal Change	Meaning Preserving Change	Minor (Meaning change)	Major (Meaning Change)
UC2.5.5 Substitute: 'Non-destroyed' -> 'Normal (Contra-lateral)'	⬭	⬭	⬭	⬭
UC2.5.5.1 i Delete: 'at least'	⬭	⬭	⬭	⬭
UC2.5.5.1 i Add: '(cortex)'	⬭	⬭	⬭	⬭
UC2.5.5.1 i Add: '(one pair just inferior to the lesser trochanter and one pair at the lowers visible region on the X-Ray)'	⬭	⬭	⬭	⬭

Case 8

Original	Revised
UC2.5.6 Identify Replacement Parameters of Destroyed Hip	*UC2.5.6 Identify Replacement Parameters of Diseased Hip*
UC2.5.6.2 Reflect CORS radial line of non-destroyed hip onto destroyed hip.	UC2.5.6.2 Reflect CORS radial line of normal hip onto diseased (ipsi-lateral) hip.
UC2.5.6.6 Draw cut-line on femur:	UC2.5.6.6 Draw initial femoral resection (cut-line) on femur:
i. draw line from top of lesser trochanter to the bottom of the head of the femur.	i. draw line from top of lesser trochanter to the bottom of the head of the femur.

8. **Rate all changes** *
Mark only one oval per row.

	Formal Change	Meaning Preserving Change	Minor (Meaning change)	Major (Meaning Change)
UC2.5.6 Substitute: 'Destroyed' -> 'Diseased'	◯	◯	◯	◯
UC2.5.6.2 Substitute: 'non-destroyed' -> 'normal'	◯	◯	◯	◯
UC2.5.6.2 Substitute: 'destroyed' -> 'diseased (ipsi-lateral)'	◯	◯	◯	◯
UC2.5.6.6 Add: 'initial femoral resection'	◯	◯	◯	◯

Case 9

Original	Revised
UC2.5.8 Select Shell	*UC2.5.8 Select Shell*
	The use enters "Select Shell" mode i.e. The AS template is added as an annotation to the X-Ray. The user is able only to perform functions related to the selection of the AS. Confirmation of completion of AS selection is required.
	UC2.5.8.4 Select Intended Replacement Type for Acetabular Shell (IRType(AS)).
UC2.5.8.1 Select Initial Acetabular Shell: Default is IRType, but can be overridden. Size biased towards preset AS diameter.	UC2.5.8.1 Select Initial Acetabular Shell: Default is IRType(AS), but can be overridden. Size biased towards preset AS diameter.
UC2.5.8.2 Change the template until Surgeon is satisfied (selecting different diameters of Shell).	UC2.5.8.2 Change the template until Surgeon is satisfied (selecting different diameters of Shell).
UC2.5.8.3 Select material for Insert.	UC2.5.8.3 Select material, internal diameter, and other attributes e.g. low profile, extended rim, of Insert.

9. **Rate all changes** *

Mark only one oval per row.

	Formal Change	Meaning Preserving Change	Minor (Meaning change)	Major (Meaning Change)
Add: "The use enters “Select Shell” mode ie. The AS template is added as an annotation to the X-ray. The user is able only to perform functions related to the selection of the AS. Confirmation of completion of AS selection is required."	◯	◯	◯	◯
UC2.5.8.4 Add: "Select Intended Replacement Type for Acetabular Shell (IRType(AS))"	◯	◯	◯	◯
UC2.5.8.1 Add: '(AS)'	◯	◯	◯	◯

Case 10

Original	Revised
UC2.5.8.3 Select Femoral Head: size and material (which must be compatible with Insert size and material).	UC2.5.8.3 Select Femoral Head: neck length, size and material (must be compatible with Insert size and material).

10. **Rate all changes** *

Mark only one oval per row.

	Formal Change	Meaning Preserving Change	Minor (Meaning change)	Major (Meaning Change)
UC2.5.9.5: Add" 'neck length'	◯	◯	◯	◯
UC2.5.9.5 Delete: 'which'	◯	◯	◯	◯

Case 11

Original	Revised
UC2.5.11.1 Adjust femur cut-line.	UC2.5.11.1 Adjust femoral neck resection line.

11. **Rate all changes** *

Mark only one oval per row.

	Formal Change	Meaning Preserving Change	Minor (Meaning change)	Major (Meaning Change)
UC2.5.11.1 Substitute: 'femur cut-line' -> 'femoral neck resection line'	◯	◯	◯	◯

Case 12

Original	Revised
UC2.7.2 On completion of the above, the surgeon will request the system to generate the orders. These orders will be saved in a format suitable for printing or email. The system creates a set of orders, one for each supplier corresponding to a component or accessory specified in the Operation Plan. All components sourced from a particular supplier will appear in that supplier's order, as will other relevant details (such as confidence levels) from the Operation Plan.	UC2.7.2 On completion of the above, the surgeon will request the system to generate the orders. These orders will be saved in a format suitable for printing or email. The system creates a set of orders, one for each supplier corresponding to a component or accessory specified in the Operation Plan. All components sourced from a particular supplier will appear in that supplier's order, as will other relevant details (such as confidence levels) from the Operation Plan. Copies of the orders are also sent to the hospital.

12. **Rate all changes** *

Mark only one oval per row.

	Formal Change	Meaning Preserving Change	Minor (Meaning change)	Major (Meaning Change)
UC2.7.2 Add: "Copies of the orders are also sent to the hospital"	◯	◯	◯	◯

Appendix C

Significant Revision Identification Annotation Guidelines

C.1 Introduction

This is the annotation guideline for the task of labelling the type of meaning change between two back-to-back versions of a text documents (Faigley and Witte 1981). *Back-to-back* versions are two revised texts by two different authors; a revision performed by an author and passed to another author and revision will further be made by this author. You will be presented with revised sentences and decide the types of meaning change, as defined and presented with examples in the section below (Section C.2). You will be further asked if you considered information beyond the scope presented to deduce the type of meaning change. The examples are presented together with the examples for the different types of meaning change.

C.2 Types of Meaning Change in Revision

The description for each type of meaning changes in revision (excluding tables, figures, citations and formulas), follow by the examples are as below:

Formal Change (F) No meaning change but changes related to Copy-Editing such as revising the Spelling, Tense, Number, and Modality, Abbreviation, Punctuation, Format, without changing the meaning of the sentence.

Original I paid a hundred dollar for the tickets to take my family to a movie.

Revised

I paid a hundred dollars for the tickets to take my family to a movie.

DID YOU CONSIDER INFORMATION BEYOND THE REVISION SCOPE PRESENTED? NO.

Meaning Preserving Change (MP) Re-phrase or Re-word to express the sentence in different style that does not change the meaning of the sentence within the context.

Original I paid a hundred dollars for the tickets to take my family to a movie.

Revised

I paid a hundred dollars to take my family to a movie.

DID YOU CONSIDER INFORMATION BEYOND THE REVISION SCOPE PRESENTED? NO.

Revised

I took my family to a movie.

I paid a hundred dollars for the tickets.

DID YOU CONSIDER INFORMATION BEYOND THE REVISION SCOPE PRESENTED? YES.

Microstructure Change (Mi) Revision that alters the meaning of words within the sentence BUT that does NOT alter the overall gist of the sentence in the greater context.

Original I paid a hundred dollars for the tickets to take my family to a movie.

Revised

I paid a hundred dollars for the tickets, with popcorns and drinks, to bring my family to a movie.

DID YOU CONSIDER INFORMATION BEYOND THE REVISION SCOPE PRESENTED? NO.

Revised It was raining heavily last night.

I paid a hundred dollars for the tickets to take my family to a movie.

DID YOU CONSIDER INFORMATION BEYOND THE REVISION SCOPE PRESENTED? YES.

Macrostructure Change (Ma) Revision to the sentence which alters the overall gist of the sentence in the greater context

Original I paid a hundred dollars for the tickets to take my family to a movie.

Revised

We decided to watch movie at home.

DID YOU CONSIDER INFORMATION BEYOND THE REVISION SCOPE PRESENTED? NO.

Revised

It does not matter how much I paid for the movie.

We did not like the movie.

DID YOU CONSIDER INFORMATION BEYOND THE REVISION SCOPE PRESENTED? YES.

C.3 Main Annotation Steps

You will be given revision text documents with the revised sentences presented, as shown Section C.4. The main annotation steps are as follow:

1. Read the revision scope presented (i.e. original and revised texts, sometimes within the revision scope presented, there are multiple edits). Deletion edits are presented in red font with strike off while additions are presented as blue coloured texts with wavy lines below, as shown in Section **??**.
2. Evaluate the type of meaning change according to Section **??**.
3. Highlight/Circle the type of meaning change in the area on the right side of the revision either as F for Formal Change, MP for Meaning Preserving Change, Mi for Microstructure Change or Ma for Macrostructure Change.
4. If you deduce the type of meaning change beyond the scope presented, highlight/circle the Yes (Y), if not, circle No (N).

Important - Within the revision scope presented, there can have multiple types of meaning changes. Label based on the "heaviest" change in this sequence:

Macrostructure Change > Microstructure Change > Meaning Preserving Change > Formal Change

C.4 Sample of the Annotation Interface

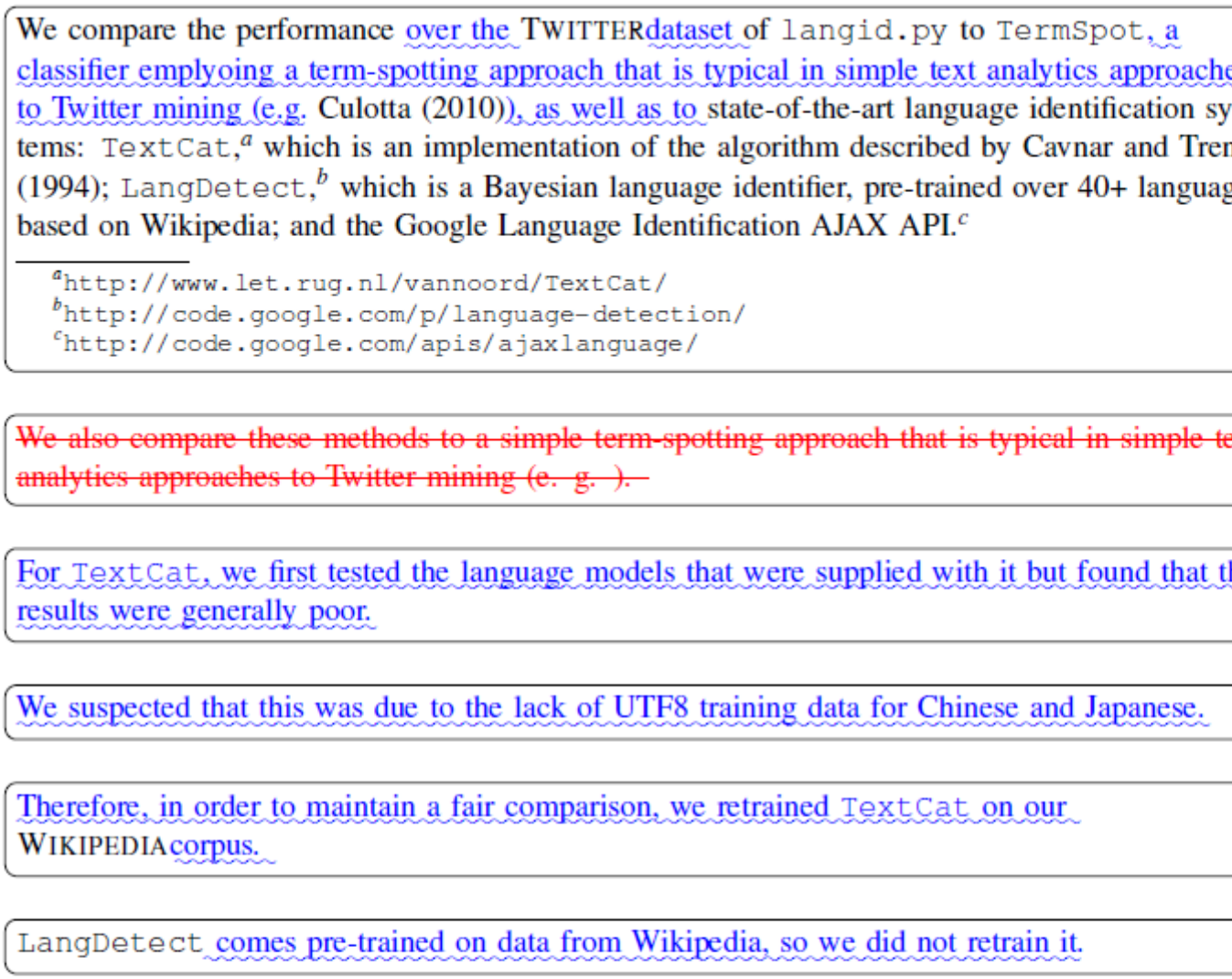


FIGURE C.1: Sample of Annotation Interface